UNIVERSITY OF OSLO
Department of Physics

Time evolution of magnetic Bloch oscillations using TEBD

Audun Skaugen

December 1, 2013

Abstract

Linear-chain ferromagnets such as $CoCl_2 \cdot 2\,H_2O$ are expected to be capable of *magnetic Bloch oscillations*. This means that domain walls in the material undergo Bloch oscillations, resulting in an oscilling magnetization. It was recently suggested to use lasers in resonance with energies from the Wannier-Zeeman ladder to provoke such oscillations. Simulating this process in the fully interacting system requires quantum many-body simulations. Since the system is one-dimensional, the method of time-evolving block decimation (TEBD) seems appropriate. We describe and implement this method and use it to simulate lasers acting on the linear chain. This results in robust oscillations in the magnetization with the expected Bloch frequency, with little sign of decoherence.

Takk

Aller først vil takke min veileder Olav Syljuåsen, som har ønsket meg alt godt siden jeg banket på døra hans for ett og et halvt år siden, og som har vist meg mye interessant fysikk.

Mamma og Pappa fortjener en stor takk for å ha tennet og næret min interesse for naturvitenskap. Min utømmelige nysjerrighet må ha vært til en viss irritasjon noen ganger, men dere gjorde alltid deres beste for å gi meg tilfredsstillende svar. Takk også til mine brødre, som har vært med på å forme meg til den personen jeg er.

Studiehverdagen på Blindern hadde vært veldig kjedelig uten alle studentforeningene, som bidrar med fester, foredrag, quiz og andre aktiviteter. En spesielt stor takk går til Studentorhesteret Biørneblæs, hvor spillegleden er større enn noe annet sted jeg vet om. Takk til alle de fantastiske personene jeg har tilbrakt tid sammen med på Blindern.

Mine medmusikanter i Føyka storband og kammerkoret Vox Humana fortjener en stor takk. De fantastiske opplevelsene det er å framføre konserter sammen med dere gir farge til tilværelsen min.

Tusen takk til Irene for all den fine tiden vi har hatt sammen, og for at du alltid lytter til mine interessante eller uinteressante tanker. Du er alltid der når jeg trenger deg.

Blindern, 30. november 2013
Audun Skaugen

Summary in Norwegian

Kvantemekanikken fører til mange uventede fenomener. Selv noe så konkret som et elektron i et krystallgitter kan oppføre seg rart. Hvis elektronet påvirkes av en *konstant* kraft skulle man vente at det havner i en likevektsposisjon mot krystallgitteret, eller begynner å vandre. Det som i stedet skjer er at elektronet begynner å svinge fram og tilbake. Utslaget til svingningen *avtar* med styrken på kraften, mens frekvensen øker.

Teorien som forutsier slike svingninger gjelder ikke bare krystallelektroner, men alle partikler som holdes på plass av en periodisk potensialkraft. Dette gjelder også såkalte *kvasipartikler*. En kvasipartikkel er en mangepartikkeleffekt som man likevel kan beskrive på samme måte som man ville beskrevet en partikkel. Et godt eksempel er i ferromagneter, hvor spinnet til lokaliserte elektroner foretrekker å peke i samme retning. Dette gjør at spinnene samler seg i såkalte domener, som er områder hvor alle spinnene er parallelle. Grensen mellom to domener kalles en domenevegg. Hvis systemet er endimensjonalt har domeneveggen en veldefinert posisjon, og kan dermed anses som en partikkel. Det viser seg at slike partikler er ypperlige kandidater til å svinge på den samme måten som vi beskrev over.

Det er ikke vanskelig å vise at ikke-vekselvirkende domenevegger i et endimensjonalt system vil svinge. I alle realistiske systemer vil det imidlertid være vekselvirkninger mellom domenevegger. For å vise at domeneveggene svinger uten å ignorere noen former for vekselvirkninger er det nødvendig å simulere kvantemekanikk med mange partikler.

Den samme grunnleggende egenskapen ved kvantemekanikk som lager disse uventede fenomenene, gjør det dessverre veldig vanskelig å regne på mange partikler. For å regne på N partikler som hver bare kan stå i to konfigurasjoner (for eksempel peke enten opp eller ned), trengs 2^N dimensjoner. Dette blir veldig fort tungt å regne på, så tilnærmingsmetoder er nødvendige for å komme videre med problemet.

En ganske ny tilnærmingsmetode med navn *time-evolving block decimation* (TEBD) egner seg spesielt godt til å studere det ovennevnte problemet. I denne oppgaven beskriver vi denne metoden og benytter den til å finne ut om Bloch-svingninger eksisterer i en modell for et spesielt magnetisk materiale med den kjemiske formelen $CoCl_2 \cdot 2\,H_2O$. Det spesielle med dette materialet er at krystallen ordner seg i rettlinjede kjeder, hvor de magnetiske vekselvirkningene mellom kjeder er mindre viktige enn innad i en kjede. Dette gjør at man med god tilnærming kan se på hver kjede for seg, slik at problemet blir endimensjonalt og mye enklere å regne på.

For å framprovosere Bloch-svingninger bruker vi lasere som er i resonans med spesielle energinivåer i krystallen. Disse resonansene er veldig smale, så for å finne de beste frekvensene prøver vi mange ulike frekvenser og analyserer resultatene. Ved hjelp av frekvensene vi finner kan vi fylle to av de ovennevnte energinivåene, som gir Bloch-svingninger. To nivåer kan fylles ved å bruke en veldig kraftig laser, men dette vil sansynligvis rive krystallen i stykker. Som et alternativ bruker vi to svakere lasere som er i resonans med hvert sitt energinivå. De resulterende Bloch svingninene er pene og robuste, og svingefrekvensen kan justeres ved å variere magnetfeltet.

Contents

Chapter 1

Introduction

It is now an established fact of physics that particles in a periodic potential undergoing a *constant* force will exhibit oscillatory motion. This counterintuitive phenomenon, known as *Bloch oscillations*, is inherently quantum mechanical, resulting from the wave-like behavior of the particle. Since their experimental observation in semiconductor superlattices (Feldmann et al., 1992; Leo et al., 1992; Waschke et al., 1993), researchers have been looking for Bloch oscillations in other kinds of systems.

One promising candidate for exhibiting Bloch oscillations is magnetic systems. As we will see, such systems contain particle-like excitations called *domain walls*, which behave as if they are subject to a periodic potential. A magnetic field will act as a constant force on the domain wall, so that one expects the wall to oscillate. However, taking interactions between the domain walls into account has proven to be difficult, due to the computational difficulties inherent in many-body quantum mechanics.

These difficulties are deeply connected to the quantum phenomenon of *entanglement*. In order to describe a particle with two states (typically labeled $\uparrow$ and $\downarrow$), we need to allow for *superpositions* of the two states. When considering several such particles interacting, it is insufficient to describe each particle separately; one must include all superpositions of the possible states of each particle. One says that the two particles can be *entangled*. Keeping track of all of this means that the time needed to simulate a system of N parts grows exponentially in N, becoming intractable even when N is relatively small.

The idea of quantum computers arises from such considerations. Intuitively, since quantum states arising from the physical laws are so difficult to compute, one could use the quantum states themselves to perform computations that would be more difficult to compute on a classical computer. However, it turns out that building such a computer is extremely difficult in practice. Since most of the physical systems that can be constructed in the laboratory do not seem to be capable of quantum computation, it was speculated that these systems do not actually process as much information as a naïve quantum analysis seems to indicate.

Starting from the argument that the crucial feature separating quantum from classical mechanics is entanglement, this should mean that realistic systems exhibit only small amounts of entanglement (in a sense which we shall make more precise below), and that this can be exploited in order to simulate the system in less than exponential time. One of the first algorithms to be developed from this line of thinking was time-evolving block decimation (TEBD; Vidal, 2003). In this thesis we attempt to make use of the TEBD algorithm in order to study the

many-body phenomenon of magnetic Bloch oscillations.

The thesis is structured as follows: We review some of the basic quantum theory that we need in chapter 2. The physical problem of laser-excited magnetic Bloch oscillations is explored in chapter 3. The numerical method of time-evolving block decimation is introduced in chapter 4, and we describe how it is implemented on the computer in chapter 5. The results are presented in chapter 6, and we conclude in chapter 7.

Chapter 2

Quantum mechanics

As described in the introduction, the method of time-evolving block decimation is motivated by ideas from quantum information theory. In order to establish a common ground for understanding this motivation, we include here a brief summary of the quantum mechanics of discrete finite systems as well as basic quantum information theory. This also allows us to emphasize the particular properties of quantum mechanics which will be of special use to us. For a more comprehensive introduction to quantum mechanics, see e.g. Sakurai (1993).

2.1 Hilbert space

States of quantum mechanical systems form a complete normed complex vector space, or a *Hilbert space* in short. Given two systems A and B with Hilbert spaces $\mathcal{H}_A$ and $\mathcal{H}_B$, we can describe the combined system by states from the *tensor product* space $\mathcal{H}_A \otimes \mathcal{H}_B$. Thus, given N identical systems described by $\mathcal{H}$, the compound system is described by $\mathcal{H}^{\otimes N}$. If $\mathcal{H}$ has a (discrete, finite) orthonormal basis $\{|i\rangle\}_{i=1}^{d}$, the compound system has an orthonormal basis consisting of the d^N states $|i_1\rangle \otimes |i_2\rangle \otimes \cdots \otimes |i_N\rangle$. This basis is called the *product basis*.

As a notational convention, we will drop the tensor-product symbol ($\otimes$) between kets when it does not lead to ambiguity, and sometimes even combine tensored kets into a single ket. Thus the following equalities hold:

$$|i_1\rangle \otimes |i_2\rangle \otimes \cdots \otimes |i_N\rangle = |i_1\rangle\,|i_2\rangle \ldots |i_N\rangle = |i_1 i_2 \ldots i_N\rangle\,. \tag{2.1}$$

2.2 Dual space

The Hilbert space $\mathcal{H}$ has a dual vector space $\mathcal{H}^\dagger$, consisting of all linear functions $\mathcal{H} \to \mathbb{C}$. Given a state $|\psi\rangle$, the linear function sending a state $|\phi\rangle$ to the inner product $\langle\psi|\phi\rangle$ is written as $\langle\psi|$ and called a *bra*. The usual axioms for an inner product for a complex vector space require that $\langle\phi|\psi\rangle = \langle\psi|\phi\rangle^*$ for any pair of vectors, and that $\langle\psi|\psi\rangle > 0$ for any nonzero vector. A particularity of quantum mechanics is that we require physical states to be *normalized*, that is $\langle\psi|\psi\rangle = 1$.

Bras and kets can be combined to form certain linear operators $\mathcal{H} \to \mathcal{H}$: Given states $|\phi\rangle\,, |\psi\rangle$, the operator $|\phi\rangle\langle\psi|$ sends a state $|\chi\rangle$ to $\langle\psi|\chi\rangle\,|\phi\rangle$. Adding such operators in the usual

way yields even more operators, and we prove in the next section that all operators can be represented in this way.

A special case of the above is *projection operators*: Given a state $|\psi\rangle$, the operator $|\psi\rangle\langle\psi|$ sends a state $|\phi\rangle$ to its projection along $|\psi\rangle$, namely $\langle\psi|\phi\rangle\,|\psi\rangle$.

2.3 Coordinate representation

By fixing an orthonormal basis $|i\rangle$ one obtains a coordinate representation of Hilbert space. Given a state $|\psi\rangle$, the coordinates ψ_i of $|\psi\rangle$ along $|i\rangle$ are defined by

$$|\psi\rangle = \sum_i \psi_i\,|i\rangle\,, \tag{2.2}$$

from which we immediately deduce

$$\langle i|\psi\rangle = \sum_j \psi_j\,\langle i|j\rangle = \psi_i. \tag{2.3}$$

This also gives a particular representation of the identity operator, given by $\hat{I} = \sum_i |i\rangle\langle i|$. Indeed,

$$\left(\sum_i |i\rangle\langle i|\right)|\psi\rangle = \sum_i |i\rangle\,(\langle i|\psi\rangle) = \sum_i \psi_i\,|i\rangle = |\psi\rangle\,. \tag{2.4}$$

Given states $|\psi\rangle\,, |\phi\rangle$ the inner product $\langle\psi|\phi\rangle$ can then be given by

$$\langle\psi|\phi\rangle = \sum_i \langle\psi|i\rangle\,\langle i|\phi\rangle = \sum_i \psi_i^*\phi_i, \tag{2.5}$$

agreeing with the usual coordinate definition of the inner product.

For an operator $\hat{O}$, we can form a matrix representing the operator by $O_{ij} = \langle i|\,\hat{O}\,|j\rangle$. The coordinate representation of a state $|\phi\rangle = \hat{O}\,|\psi\rangle$ transformed by $\hat{O}$ can then be given in terms of this matrix by

$$\phi_i = \langle i|\,\hat{O}\,|\psi\rangle = \sum_j \langle i|\,\hat{O}\,|j\rangle\,\langle j|\psi\rangle = \sum_j O_{ij}\psi_j, \tag{2.6}$$

agreeing with the usual matrix-vector product. Similarly, for a composition $\hat{O} = \hat{V}\hat{W}$ of two operators, the matrix elements are given by

$$O_{ij} = \langle i|\,\hat{O}\,|j\rangle = \sum_k \langle i|\,\hat{V}\,|k\rangle\langle k|\,\hat{W}\,|j\rangle = \sum_k V_{ik}W_{kj}, \tag{2.7}$$

agreeing with the usual matrix product. To regain an operator from the matrix form, we can use the operator basis given by $\{|i\rangle\langle j|\}_{i,j=1}^d$, giving

$$\sum_{ij} O_{ij}\,|i\rangle\langle j| = \sum_{ij} |i\rangle\langle i|\,\hat{O}\,|j\rangle\langle j| = \hat{O}. \tag{2.8}$$

This also shows that all operators can be written in this basis.

2.3.1 Tensor products and index combination

It will be useful to consider the coordinate representations of tensor products. A state $|\psi\rangle$ in the tensor product space $\mathcal{H}_1 \otimes \mathcal{H}_2$, with $\mathcal{H}_1$ spanned by basis $\{|i_1\rangle\}_{i=1}^{d_1}$ and $\mathcal{H}_2$ spanned by basis $\{|j_2\rangle\}_{j=1}^{d_2}$, has expansion coefficients ψ_{ij} defined by

$$|\psi\rangle = \sum_{i=1}^{d_1}\sum_{j=1}^{d_2} \psi_{ij}\,|i_1 j_2\rangle\,. \tag{2.9}$$

From this we can see that $\psi_{ij} = \langle i_1 j_2|\psi\rangle$. These expansion coefficients can be regarded as a $d_1 \times d_2$ matrix, but we should also be able to regard them as a $d_1 d_2$-dimensional coordinate vector.

To this end we introduce an index-combining notation (ij), which means that the pair of indices is combined into a single compound index that ranges over all pairs. This can for instance be done with $(ij) = d_2 i + j$. Given this definition, the two indices can be extracted from a compound index k by $i = \lfloor k/d_2 \rfloor$ and $j = k \bmod d_2$, e.g. by the result and remainder of dividing k by d_2. Similarly, using three indices one can define $(ijk) = d_2 d_3 i + d_3 j + k$, with d_3 the range of the k index. With such definitions we can define the coordinate vector by $\psi'_{(ij)} = \psi_{ij}$.

For linear operators similar considerations apply. Given an operator $\hat{O}$ acting on $\mathcal{H}_1 \otimes \mathcal{H}_2$, the matrix elements of $\hat{O}$ are given by $O^{ij}_{kl} = \langle ij|\,\hat{O}\,|kl\rangle$. These coefficients can be regarded as a rank-4 tensor, but it will be useful to regard them as a matrix as well. Thus we define a $d_1 d_2 \times d_1 d_2$ matrix O with coefficients $O_{(ij),(kl)} = O^{ij}_{kl}$. From this we can for instance find the matrix describing a simple tensor product operator $\hat{O} = \hat{A} \otimes \hat{B}$ by

$$O_{(ij),(kl)} = \langle ij|\,(\hat{A}\otimes\hat{B})\,|kl\rangle = \langle i|\,\hat{A}\,|k\rangle\,\langle j|\,\hat{B}\,|l\rangle = A_{ik}B_{jl}. \tag{2.10}$$

This agrees with the usual Kronecker product of matrices.

2.4 Observables

Observables of the system are described by hermitean linear operators $\hat{O}$ acting on Hilbert space. Hermiticity means that the matrix $O_{ij} = \langle i|\,\hat{O}\,|j\rangle$ is hermitean and can be diagonalized, yielding a complete orthonormal basis of eigenkets $|o_i\rangle$ and associated real eigenvalues o_i, with $\hat{O}\,|o_i\rangle = o_i\,|o_i\rangle$. Using the identity operator $\sum_i |o_i\rangle\langle o_i|$, we obtain an elegant representation of $\hat{O}$:

$$\hat{O} = \sum_i \hat{O}\,|o_i\rangle\langle o_i| = \sum_i o_i\,|o_i\rangle\langle o_i|\,. \tag{2.11}$$

This is called the *spectral representation* of $\hat{O}$.

The spectral representation is useful for calculating the result of measuring the observable $\hat{O}$: An observable acting on one of its eigenstates will produce the same state multiplied by the eigenvalue for that state, while an observable acting on a generic state will produce a linear combination of the possible measurement outcomes, each multiplied by the associated eigenvalue. An actual measurement will choose one eigenstate in this linear combination, with the

probability of obtaining $|o_i\rangle$ given by $|\langle o_i|\psi\rangle|^2$, the modulus square of the projection of the state along the eigenstate. When the measurement is finished, the state will be in this eigenstate: $|\psi\rangle = |o_i\rangle$. This assures that repeated measurements of the same observable yields the same value.

Some care needs to be taken when applying this rule to *degenerate* observables. Observables are degenerate when two or more eigenvalues are equal. Since we can not distinguish between two states with the same eigenvalue, the probability of obtaining this value is the sum of the probabilities for each of the states with the given value. After the measurement is finished, the state is in a superposition involving all of the degenerate eigenstates.

The rule for obtaining measurement results means that if we multiply a state with a complex phase $e^{i\theta}$, all measurement outcomes will yield the same result. Thus, a global change of phase carries no physical content.

The expectation value for the outcome of a measurement is given by the usual statistical formula,

$$\left\langle \hat{O} \right\rangle_\psi = \sum_i o_i |\langle o_i|\psi\rangle|^2 = \sum_i o_i \langle\psi|o_i\rangle \langle o_i|\psi\rangle = \langle\psi| \hat{O} |\psi\rangle . \tag{2.12}$$

Each physical system has a special observable called the *Hamiltonian*, which measures the energy of a given state, as well as generating its time-development. Given a time-dependent state $|\psi(t)\rangle$, its time-evolution obeys the *Schrödinger equation*

$$i\hbar \frac{\mathrm{d}|\psi(t)\rangle}{\mathrm{d}t} = \hat{H}(t)|\psi(t)\rangle . \tag{2.13}$$

If the Hamiltonian is time-independent, this equation has the solution

$$|\psi(t)\rangle = e^{-\frac{i}{\hbar}\hat{H}(t-t_0)}|\psi(t_0)\rangle = \hat{U}(t-t_0)|\psi(t_0)\rangle , \tag{2.14}$$

where the exponential of an operator is defined by the series expansion

$$e^{\hat{A}} = 1 + \hat{A} + \frac{1}{2!}\hat{A}^2 + \frac{1}{3!}\hat{A}^3 + \dots \tag{2.15}$$

Using the spectral representation $\hat{H} = \sum_n E_n |n\rangle\langle n|$, we can calculate powers easily:

$$\hat{H}^2 = \sum_{mn} E_m E_n |m\rangle \langle m|n\rangle \langle n| = \sum_n E_n^2 |n\rangle\langle n| . \tag{2.16}$$

This lets us calculate the exponential as

$$e^{x\hat{H}} = \sum_n \left(1 + x\hat{E}_n + \frac{1}{2!}x^2 E_n^2 + \frac{1}{3!}x^3 E_n^3 \dots\right) |n\rangle\langle n| = \sum_n e^{xE_n} |n\rangle\langle n| , \tag{2.17}$$

giving the convenient expression

$$\hat{U}(t-t_0) = \sum_n e^{-\frac{i}{\hbar}E_n(t-t_0)} |n\rangle\langle n| . \tag{2.18}$$

For time-dependent Hamiltonians, solving the Schrödinger equation is usually more compli-

cated. However, we will later make use of a small trick: If one subdivides the time into small enough timesteps, the Hamiltonian will be approximately constant on each timestep. From one such timestep to another, the Schrödinger equation has the solution

$$|\psi(t+\delta t)\rangle \approx \hat{U}(t,\delta t)\,|\psi(t)\rangle = e^{-\frac{i}{\hbar}\hat{H}(t)\delta t}\,|\psi(t)\rangle\,. \tag{2.19}$$

2.5 Density operators and entanglement

Consider a compound system described by the Hilbert space $\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B$, with $\mathcal{H}_A$ spanned by a basis $\{|i_A\rangle\}_{i=1}^{d_A}$ and $\mathcal{H}_B$ spanned by a basis $\{|j_B\rangle\}_{j=1}^{d_B}$. Any state $|\psi\rangle$ in $\mathcal{H}$ can then be given by a double sum over the basis states:

$$|\psi\rangle = \sum_{i=1}^{d_A}\sum_{j=1}^{d_B} c_{ij}\,|i_A\rangle\,|j_B\rangle\,. \tag{2.20}$$

It would be nice if it were possible to describe each subsystem separately, by assigning states $|\psi_A\rangle\,, |\psi_B\rangle$ to each. We would then require the tensor product of the two states to yield the same description as the one we would find by considering the complete system:

$$|\psi_A\rangle\,|\psi_B\rangle = |\psi\rangle\,. \tag{2.21}$$

Expanding the subsystem states as $|\psi_A\rangle = \sum_i a_i\,|i_A\rangle\,, |\psi_B\rangle = \sum_j b_j\,|j_B\rangle$, we obtain

$$\sum_{ij} a_i b_i\,|i_A\rangle\,|j_B\rangle = \sum_{ij} c_{ij}\,|i_A\rangle\,|j_B\rangle\,, \quad \text{ie.} \quad c_{ij} = a_i b_j. \tag{2.22}$$

It is clear that this can not be satisfied in general. Thus quantum states do not admit complete descriptions for each subsystem. This phenomenon is known as *entanglement*, and is the crucial property separating quantum from classical mechanics.

In order to describe a subsystem by itself in the presence of entanglement, a generalization of the quantum state has to be employed. The resulting object is called a *mixed state*, and is described by a *density operator*. This operator can be represented by a $d \times d$ matrix, with d the dimension of the subsystem described; the matrix is called a *density matrix*.

The density operator is formed from the complete state $|\psi\rangle$ by forming the projection operator $|\psi\rangle\langle\psi|$, and then *tracing out* the subsystem we are not interested in. In the example above, to describe subsystem A, we form the operator

$$\begin{aligned}\hat{\rho}_A = \mathrm{Tr}_B\,|\psi\rangle\langle\psi| &= \sum_j \langle j_B|\psi\rangle\,\langle\psi|j_B\rangle \\ &= \sum_j \sum_{ii'kk'} c_{ik}c^*_{i'k'}\,\langle j_B|k_B\rangle\,\langle k'_B|j_B\rangle\,|i_A\rangle\langle i'_A| \\ &= \sum_{ijk} c_{ij}c^*_{kj}\,|i_A\rangle\langle k_A|\,. \end{aligned} \tag{2.23}$$

This operator is hermitean, and the diagonal elements $\langle i_A|\,\hat{\rho}_A\,|i_A\rangle = \sum_j |c_{ij}|^2$ are real and

positive. Since the c_{ij} are expansion coefficients, the diagonal sums to one:

$$\operatorname{Tr}\hat{\rho} = \sum_{ij} |c_{ij}|^2 = 1, \tag{2.24}$$

which means that the diagonal elements can be interpreted as probabilities.

In order to interpret these probabilities, consider trying to measure which basis state the A subsystem is in. To do this we form the operator $\hat{O} = \sum_i i\,|i_A\rangle\langle i_A|$, so that the result of measuring $\hat{O}$ is a tag indicating which state was found. We then extend this operator to one acting on the complete system by using the identity on the B subsystem, so that the complete observable is $\hat{O}\otimes I$. Accounting for the degeneracy of the $\mathcal{H}_B$ identity operator, the probability of measuring the i tag is

$$p_i = \sum_j |\,\langle i_A j_B|\psi\rangle\,|^2 = \sum_j |c_{ij}|^2 = \langle i_A|\,\hat{\rho}_A\,|i_A\rangle\,. \tag{2.25}$$

Thus the diagonal elements of $\hat{\rho}_A$ describe the probabilities for measuring the A subsystem to be in the corresponding basis state.

This leads to an equivalent interpretation of the mixed state as representing incomplete information: Because we restrict attention to a subsystem of the complete system, we no longer have the means to describe the state accurately. We have to resort to a statistical mixture of pure states (hence *mixed state*). Note, however, that many different sets of pure states can be mixed together to form the same mixed state, so the interpretation as a mixture of pure states is not unique.

To further demonstrate the utility of the density operator, consider a more general observable $\hat{O}$ for the subsystem $\mathcal{H}_A$. Once again we tensor it with the identity for $\mathcal{H}_B$, but now we calculate the expectation value:

$$\begin{aligned}\left\langle \hat{O} \right\rangle &= \langle\psi|\,\hat{O}\otimes\hat{I}\,|\psi\rangle = \sum_{iji'j'} c_{ij}c^*_{i'j'}\,\langle i'_A|\,\hat{O}\,|i_A\rangle\,\langle j'_B|j_B\rangle = \sum_{ijk} c_{ij}c^*_{kj}\,\langle k_A|\,\hat{O}\,|i_A\rangle \\ &= \operatorname{Tr}\left(\sum_{ijk} c_{ij}c^*_{kj}\,|i_A\rangle\langle k_A|\,\hat{O}\right) = \operatorname{Tr}\left(\hat{\rho}\hat{O}\right).\end{aligned} \tag{2.26}$$

Here we have made use of the fact that for any states $|\phi\rangle\,, |\chi\rangle$

$$\langle\phi|\chi\rangle = \sum_i \langle\phi|i\rangle\,\langle i|\chi\rangle = \sum_i \langle i|\chi\rangle\,\langle\phi|i\rangle = \operatorname{Tr}\left(\,|\chi\rangle\langle\phi|\,\right). \tag{2.27}$$

By similar techniques one can compute all physical properties of the subsystem using the density operator.

There is a convenient way to measure the degree to which two subsystems are entangled, called the entanglement entropy. This is defined using the density operator for one of the subsystems:

$$S = -\operatorname{Tr}(\hat{\rho}\log\hat{\rho}) = -\sum_i p_i \log p_i, \tag{2.28}$$

where the logarithm is usually taken in base 2.

2.6 The Schmidt decomposition

When considering the quantum information-theoretical properties of a state, a particular decomposition in terms of subsystem kets is often useful. The *Schmidt decomposition* is a representation of $|\psi\rangle$ in terms of a *single sum*, in terms of a special orthonormal basis *particular to* $|\psi\rangle$:

$$|\psi\rangle = \sum_{i=1}^{\chi} \lambda_i \left|\Phi_i^{[A]}\right\rangle \left|\Phi_i^{[B]}\right\rangle, \tag{2.29}$$

where the *Schmidt rank* $\chi \leq \min\{d_A, d_B\}$ is bounded by the dimensinalities of the subsystems. The *Schmidt coefficients* λ_i can be chosen to be real and positive, decreasing with i.

Mathematically, the Schmidt decomposition is equivalent to the Singular Value decomposition (Lay, 2006, Chap. 7.4). Regarding the coefficients c_{ij} in equation (2.20) as a $d_A \times d_B$ matrix C, the singular value decomposition gives unitary matrices L, R and a diagonal $d_A \times d_B$ matrix Σ containing the decreasing singular values σ_k, such that $C = L\Sigma R^\dagger$. Setting χ to be the number of nonzero singular values, we find

$$c_{ij} = \sum_{k=1}^{\chi} L_{ik}\sigma_k (R_{jk})^*. \tag{2.30}$$

Inserting this into (2.20) gives

$$\begin{aligned} |\psi\rangle &= \sum_{ijk} L_{ik}\sigma_k (R_{jk})^* |i_A\rangle |j_B\rangle = \sum_{k=1}^{\chi} \sigma_k \left(\sum_{i=1}^{d_A} L_{ik} |i_A\rangle\right) \left(\sum_{j=1}^{d_B} (R_{jk})^* |j_B\rangle\right) \\ &= \sum_{k=1}^{\chi} \lambda_k \left|\Phi_k^{[A]}\right\rangle \left|\Phi_k^{[B]}\right\rangle, \end{aligned} \tag{2.31}$$

yielding the desired decomposition.

Given a Schmidt decomposition, the reduced density operator for each subsystem is diagonal in the Schmidt basis, giving a simple entanglement entropy:

$$\hat{\rho}_A = \sum_i \lambda_i^2 \left|\Phi_i^{[A]}\right\rangle \left\langle\Phi_i^{[A]}\right|, \quad S = -\sum_i \lambda_i^2 \log \lambda_i^2. \tag{2.32}$$

Moreover, this entropy is upper-bounded by $\log \chi$.

It will be useful to consider how one can force the entropy to be small, given a particular singular value decomposition. The function $p^2 \log p^2$ is plotted in figure 2.1, for $0 \leq p \leq 1$. It is close to zero only near the end points 0 and 1, so we will need the Schmidt coefficients to lie in one of these regions. However, because of the normalization requirement, only one of the coefficients can be close to 1. Thus, for a small entanglement all of the coefficients except the first must be small.

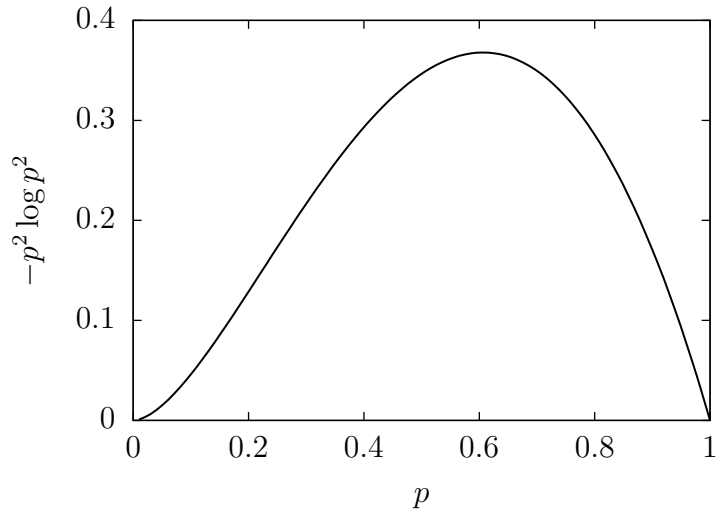


Figure 2.1: The function $-p^2 \log p^2$ for $0 \leq p \leq 1$, illustrating how the individual Schmidt coefficients contribute to total entanglement entropy.

2.7 Suzuki-Trotter decomposition

As we have seen, the time-development operator can becomputed as the exponential $e^{-i\hat{H}\delta t}$. The matrix representing the Hamiltonian $\hat{H}$ is prohibitively big for large systems, so we will need to represent it in a different way While the details of this representation are left for later, we will make use of the fact that the Hamiltonian can be decomposed in to a sum of simpler terms $\hat{H} = \sum_n \hat{H}_n$, where each of the terms can be represented efficiently. Thus, we need to calculate exponentials of a sum of terms without going by the complete sum.

Here we will consider a simpler problem, namely calculating $e^{\lambda\hat{A}+\lambda\hat{B}}$. If we could use the multiplicative property of the exponential function on numbers, we would obtain $e^{\lambda\hat{A}}e^{\lambda\hat{B}}$, where each of the exponentials only involve the simpler terms $\hat{A}$ and $\hat{B}$. This is actually valid when the operators commute, as can be shown e.g. by manipulating the series expansions. For noncommuting operators we can not expect such a manipulation to be valid, as we can use the exponential of the sum to change the order of the exponentials: $e^{\hat{A}}e^{\hat{B}} = e^{\hat{B}+\hat{A}} = e^{\hat{B}}e^{\hat{A}}$. However, the Baker-Campbell-Hausdorff formula tells us that it is approximately correct for small λ:

$$e^{\lambda\hat{A}}e^{\lambda\hat{B}} = e^{\lambda\hat{A}+\lambda\hat{B}+\frac{1}{2}\lambda^2[\hat{A},\hat{B}]+O(\lambda^3)}. \tag{2.33}$$

This is not as useful for us as it could have been, for two reasons. Firstly, we would like a formula going the other way, namely from $e^{\lambda\hat{A}+\lambda\hat{B}}$ to something involving the individual exponentials. Secondly, higher-order terms appear in the exponent, when it would be more convenient to have them as separate terms in order to derive an error estimate.

Luckily, such an expression exists and is called the Suzuki-Trotter expansion (Suzuki, 1990). It is actually a family of expansions of the form $e^{\lambda\hat{A}+\lambda\hat{B}} = f_m\left(e^{\lambda\hat{A}}, e^{\lambda\hat{B}}\right)$, valid to order m in λ. The simplest is the first-order, namely $f_1(\hat{U},\hat{V}) = \hat{U}\hat{V}$, corresponding to the naïve manipulation we performed above. The second-order decomposition is just a more symmetric version, with $f_2(\hat{U},\hat{V}) = \hat{U}^{\frac{1}{2}}\hat{V}\hat{U}^{\frac{1}{2}}$. For higher-order expansions there are many possible choices with

Table 2.1: Decimal expansions for the irrational numbers used in one of the third-order Trotter decompositions.

$$
\begin{aligned}
a_1 &= \quad 0.4515255132085857234095788 20 \\
a_2 &= -0.6308809540300025007916636 63 \\
a_3 &= -1.1367109252139957147282065 49 \\
a_4 &= -1.2191173924525839389294490 32
\end{aligned}
$$

different constant factors in the errors. Finding the optimal higher-order expansions is therefore an optimization problem.

This problem was considered in Sornborger and Stewart (1999), where the authors found optimal expressions for the third- and fourth-order decompositions. They considered two cases for each order: Restricting the powers of operators to be integers, and allowing them to be irrational numbers. Decompositions are written in a notation (αa), where α is a sign indicating the direction in time, and a is a positive exponent of the operators. Thus, in our context, (-2) means $\hat{U}^{-2}\hat{V}^{-2}$. Adding a superscript T means that the order of operators should be reversed, so e.g. $(1)^T$ means $\hat{V}\hat{U}$. Many optimal decompositions were found, but in this thesis we chose the following two third-order decompositions:

$$
f_{\text{3int}} = (1)^T(1)(1)(1)(1)^T(-2)^T(1)(1)(1), \quad f_{\text{3irr}} = (a_1)(-a_2)^T(-a_3)^T(a_4), \tag{2.34}
$$

where the a_i denote certain irrational given in table 2.1.

Chapter 3

Physical problem

In this chapter we introduce the physical problems central to the thesis. We start by reviewing the general phenomenon of Bloch oscillations. We then investigate the material that we will be interested in modeling, and show why magnetic Bloch oscillations are expected to be present there. Finally, we show how lasers in resonance with the energy levels characteristic of Bloch oscillations can populate the corresponding states, in a process known as Rabi oscillation.

3.1 Bloch oscillations

Bloch oscillations is a somewhat counter-intuitive quantum phenomenon where a particle in a periodic potential, subjected to a *constant force*, exhibits periodic motion (Bloch, 1929). While the simplest derivation of such behavior is quite straightforward (see below), it took until the early nineties before it was observed experimentally. Because of this, there was a long controversy about whether the phenomenon exists at all. This was not helped by the fact that the simplest derivation is semiclassical, allowing several openings for error.

In this section we give the semiclassical derivation of Bloch oscillations and sketch the criticisms this and other approaches have received. A more comprehensive review of the controversy is given in the introduction to Krieger and Iafrate (1986).

3.1.1 Semiclassical derivation

An intuitive explanation of Bloch oscillations can be given using the concept of *umklapp* processes. A particle in a periodic potential can be modeled as a quasiparticle with momentum confined to the first *Brillouin zone*, which is the fundamental cell of the reciprocal lattice (see Kittel, 2005). When the particle is subjected to an external force, its momentum increases until it crosses the boundary of the Brillouin zone. At this point the particle undergoes an umklapp process, reflecting the momentum from p to $-p$. This causes the particle to move in the opposite direction from before. The process repeats when the momentum has again been increased to p, leading to oscillatory motion.

To make this rigorous, consider an electron in a periodic one-dimensional potential $V(x)$, with period a, so that $V(x+a) = V(x)$ for all x. In such a system, electrons are best described by waves with a certain wavenumber k. Waves can exist in certain *energy bands*, i.e. allowed sets of energies. Within such a band, the energy of a wave is given by a *dispersion relation* $\epsilon(k)$.

Assume the electron starts as a wave-packet with wavenumber k. If the electron stays in a single band, Bloch showed that such a wave subjected to an external force F will satisfy an equation of motion analogous to Newton's second law (Kittel, 2005, chap. 7):

$$\frac{\mathrm{d}p}{\mathrm{d}t} = \hbar\frac{\mathrm{d}k}{\mathrm{d}t} = F. \tag{3.1}$$

Setting up a constant external field E gives a force $F = eE$, so that $k(t) = eE(t - t_0)/\hbar$. The group velocity of the wave packet is given by the derivative of the dispersion relation:

$$v_g = \frac{1}{\hbar}\frac{\mathrm{d}\epsilon}{\mathrm{d}k}. \tag{3.2}$$

For concreteness, we choose a simple dispersion relation with typical properties, namely $\epsilon = \epsilon_0(1 - \cos ka)$. With this choice,

$$v_g = \frac{\epsilon_0 a}{\hbar}\sin ka = \frac{\epsilon_0 a}{\hbar}\sin\frac{eE(t - t_0)a}{\hbar}. \tag{3.3}$$

Integrating, we find that the position of the wavepacket satisfies

$$x(t) = \text{const.} - \frac{\epsilon_0}{eE}\cos\omega_B(t - t_0), \tag{3.4}$$

where the *Bloch frequency* is $\omega_B = eEa/\hbar$. This frequency increases with the applied force. The amplitude, by contrast, *decreases*.

3.1.2 Quantum derivation

The above derivation is subject to several lines of criticism. The most obvious objection is that the semiclassical approach inherently neglects tunneling between energy bands. This can be alleviated by performing a fully quantum analysis of the system where the constant force is present from the beginning. The first problem that arises for such an analysis is that the translational invariance is broken by the linear potential necessary to produce a constant force, which makes it difficult to properly analyze the periodic potential.

Despite this, some early progress was made by Wannier (1960), who derived an effective Hamiltonian for the system and showed that one can modify the energy bands from the zero-field solution in such a way that no tunneling happens between the new bands. From this he found an equidistant ladder of energy bands, later called the Wannier-Stark ladder, with energy spacing ω_B. As such an equidistant ladder will always give rise to harmonic oscillation of some observable with frequency ω_B (see below), this was taken as proof that Bloch oscillations exist.

A number of objections were raised against this approach. One line of criticism is related to the unsuitability of periodic boundary conditions when the potential $V(x) = V_0 x$ is explicitly aperiodic. Another line starts from other arguments that show that the energy spectrum is in fact continuous, so that the Wannier-Stark ladder can at most represent metastable states.

The debate was only settled when Bloch oscillations were first observed in semiconductor superlattices (Feldmann et al., 1992; Leo et al., 1992; Waschke et al., 1993). Since then Bloch oscillations have been found in many different systems. One particular candidate which we

shall explore is domain walls in ferromagnetic systems.

3.1.3 Ladders and harmonic motion

In this section we will show how equidistant energy states give rise to harmonic motion. This involves considering the usual harmonic oscillator from the opposite perspective compared to the usual one: We start with the energy spectrum and calculate certain expectation values, which are seen to oscillate harmonically with the expected frequency.

Consider then a general Hamiltonian for an equidistant energy spectrum with energy spacing $\hbar\omega_B$. By shifting the energy so that the lowest rung of the ladder has zero energy, the Hamiltonian can be written as $\hat{H} = \hbar\omega_B \sum_{nl} n \left|n, l\right\rangle\!\left\langle n, l\right|$, where the l quantum number accounts for any degeneracy of the energy states. Introducing creation and annihilation operators by

$$\hat{a}^\dagger \left|n, l\right\rangle = \sqrt{n+1}\left|n+1, l\right\rangle, \quad \hat{a}\left|n, l\right\rangle = \sqrt{n}\left|n-1, l\right\rangle, \tag{3.5}$$

we find that $\hat{a}^\dagger\hat{a}\left|n, l\right\rangle = n\left|n, l\right\rangle$, so that the Hamiltonian can be rewritten as $\hat{H} = \hbar\omega_B\hat{a}^\dagger\hat{a}$. For the commutator between the ladder operators it suffices to consider basis states,

$$[\hat{a}, \hat{a}^\dagger]\left|n, l\right\rangle = (n+1)\left|n, l\right\rangle - n\left|n, l\right\rangle = \left|n, l\right\rangle, \tag{3.6}$$

so that $[\hat{a}, \hat{a}^\dagger] = 1$. To find a candidate observable exhibiting oscillating behavior, we take inspiration from the usual definitions in the harmonic oscillator, and define $\hat{x} = a^\dagger + a$. An energy eigenstate will exhibit no time-dependence, so we consider a general superposition $\left|\psi\right\rangle = \sum_n c_n \left|n\right\rangle$ of ladder states.

The time-dependent expectation value of the $\hat{x}$ operator is then found by acting on the state $\left|\psi\right\rangle$ with the time-development operator $e^{-i\omega_B\hat{a}^\dagger\hat{a}t}$ and using the definition of the expectation value to find

$$\langle\hat{x}\rangle(t) = \left\langle\psi\right| e^{i\omega_B\hat{a}^\dagger\hat{a}t}(\hat{a}^\dagger + \hat{a})e^{-i\omega_B\hat{a}^\dagger\hat{a}t}\left|\psi\right\rangle = 2\,\mathrm{Re}\left[\left\langle\psi\right| e^{i\omega_B\hat{a}^\dagger\hat{a}t}\hat{a}^\dagger e^{-i\omega_B t\hat{a}^\dagger\hat{a}t}\left|\psi\right\rangle\right]. \tag{3.7}$$

To simplify this expression we make use of the Baker-Hausdorff lemma (Sakurai, 1993, p. 96), which says that for operators $\hat{G}, \hat{A}$ and scalar λ we have

$$e^{i\lambda\hat{G}}\hat{A}\hat{e}^{-i\lambda\hat{G}} = \hat{A} + i\lambda[\hat{G}, \hat{A}] + \frac{(i\lambda)^2}{2!}[\hat{G}, [\hat{G}, \hat{A}]] + \frac{(i\lambda)^3}{3!}[\hat{G}, [\hat{G}, [\hat{G}, \hat{A}]]] + \ldots \tag{3.8}$$

When we insert equation (3.7) into this formula we need the commutator

$$[\hat{a}^\dagger\hat{a}, \hat{a}^\dagger] = \hat{a}^\dagger[\hat{a}, \hat{a}^\dagger] + [\hat{a}^\dagger, \hat{a}^\dagger]\hat{a} = \hat{a}^\dagger. \tag{3.9}$$

Since this commutator is equal to the operator we started with, all of the higher-order commutators are equal to $\hat{a}^\dagger$. The operator can therefore be factored out of the series:

$$e^{i\omega_B\hat{a}^\dagger\hat{a}t}\hat{a}^\dagger e^{-i\omega_B\hat{a}^\dagger\hat{a}} = \hat{a}^\dagger + i\omega_B t\hat{a}^\dagger + \frac{(i\omega_B t)^2}{2!}\hat{a}^\dagger + \frac{(i\omega_B t)^3}{3!}\hat{a}^\dagger + \ldots = e^{i\omega_B t}\hat{a}^\dagger. \tag{3.10}$$

The time-dependent expectation value is then given by

$$\begin{aligned}\langle \hat{x} \rangle (t) &= 2 \operatorname{Re} \left[e^{i\omega_B t} \langle \psi | \hat{a}^\dagger | \psi \rangle \right] = 2 \operatorname{Re} \left(e^{i\omega_B t} \sum_n \sqrt{n} c_n^* c_{n-1} \right) \\ &= 2a \cos \omega_B t - 2b \sin \omega_B t, \end{aligned} \tag{3.11}$$

where $a + ib = \sum_n \sqrt{n} c_n^* c_{n-1}$. Thus the expectation value will oscillate with frequency ω_B as long as any pair of neighboring states in the ladder is occupied, so that the sum has a nonzero term.

3.2 The material

The material of interest, with chemical formula $CoCl_2 \cdot 2\,H_2O$, is a salt where the cobalt atoms give away their two $4s$ electrons to the chlorine atoms, leaving an outer shell of seven $3d$ electrons. The anisotropy brought about by the crystal structure lifts the degeneracy of the d shell in such a way that pairs of states are left degenerate. When such a shell is filled with seven electrons the lower six states are filled up, while the top pair contains only a single electron. The top two states then behave effectively as a two-level system, modeled as a spin-$\frac{1}{2}$ particle. The exchange interaction with top-shell electrons in nearby cobalt ions is then modeled as an energy depending on these spins.

Another effect of the crystal structure is to order the cobalt ions in linear chains screened by water molecules, so that the interaction within a given chain is much stronger than the interaction between chains (Torrance and Tinkham, 1969). This means that it is a good approximation to model the inter-chain interaction as a mean field, leaving noninteracting chains that can be considered separately.

Since the exchange interactions decay quickly with distance, another good approximation is to consider interactions only between neighboring particles. This leads us to the Heisenberg model, which describes a Hamiltonian of the form

$$\begin{aligned}\hat{H} &= -J_z \sum_{j=1}^{N-1} \hat{S}_j^z \hat{S}_{j+1}^z - J_x \sum_{j=1}^{N-1} \hat{S}_j^x \hat{S}_{j+1}^x - J_y \sum_{j=1}^{N-1} \hat{S}_j^y \hat{S}_{j+1}^y \\ &= -J_z \sum_{j=1}^{N-1} \hat{S}_j^z \hat{S}_{j+1}^z - \sum_{j=1}^{N-1} \left[J_a \left(\hat{S}_j^+ \hat{S}_{j+1}^+ + \hat{S}_j^- \hat{S}_{j+1}^- \right) + J_\perp \left(\hat{S}_j^+ S_{j+1}^- + \hat{S}_j^- S_{j+1}^+ \right) \right] \\ &= \hat{H}_z + \hat{H}_a + \hat{H}_\perp, \end{aligned} \tag{3.12}$$

where we have introduced the raising and lowering spin operators $\hat{S}^\pm = \hat{S}_x \pm i\hat{S}_y$ and defined $J_a = (J_x - J_y)/4$, $J_\perp = (J_x + J_y)/4$. Results from far-infrared spectroscopy show that this model accounts for the low-energy behavior of the material to a remarkable accuracy (Torrance and Tinkham, 1969).

The coupling constants are given in table 3.1. All coupling constants are positive, which means that the chain has ferromagnetic behavior. Furthermore, the interaction in the z direction is much stronger than in the other directions, i.e. we have $J_z \gg J_x$ and $J_z \gg J_y$. This means

Table 3.1: Coupling constants describing the Heisenberg model for $CoCl_2 \cdot 2\,H_2O$

Constant	Value
J_z	36.5 K
J_a	3.8 K
$J_\perp$	5.43 K

$$|j,1\rangle = \left|\underbrace{\uparrow\uparrow\ldots\uparrow}_{j}\underbrace{\downarrow\downarrow\ldots\downarrow}_{N-j}\right\rangle \qquad |j,-1\rangle = \left|\underbrace{\downarrow\downarrow\ldots\downarrow}_{j}\underbrace{\uparrow\uparrow\ldots\uparrow}_{N-j}\right\rangle$$

Figure 3.1: Illustration of how the $|j,Q\rangle$ states are formed.

that the system will predominantly act as an *Ising model* (meaning that the only coupling is in the z direction), with small corrections given by the $\hat{H}_a, \hat{H}_\perp$ terms.

3.2.1 Ising model behavior

The Ising model described by $\hat{H}_z$ gives a very simple behavior of the system. As the Hamiltonian consists only of $\hat{S}_z$ operators, the eigenstates are product states of $\hat{S}_z$ eigenkets. The energy of a pair of spins is given by a single term in the sum, which gives $-J_z/4$ if the spins are parallel, and $J_z/4$ if they are antiparallel. This means that both the *ferromagnetic states*, where all the spins are parallel, are ground states.

The lowest-lying excitations will then be states where the spins are separated into two *domains*, with parallel spins in each domain. The boundary between the domains is called a *domain wall*. The energy cost of the domain wall is the difference between the energy $J_z/4$ of the wall and the energy $-J_z/4$ if it had been in the parallel ground state, yielding an energy gap of $E_g = J_z/2$.

It is clear that the position of this domain wall does not matter for the energy of the excitation. The excited state is therefore highly degenerate.

3.2.2 Effect of the other couplings

The $\hat{H}_a$ and $\hat{H}_\perp$ terms do not commute with $\hat{H}_z$, so when we take these into account the simple product state picture we described above breaks down. However, since the additional coupling constants are small, the true ground state and low-lying excited states should not be far from those we considered above. Using this, Kyriakidis and Loss (1998) introduced the *single domain-wall* approximation. This approximation amounts to only considering states with at most one domain wall. From this approximation one can derive an effective Hamiltonian which can be diagonalized to obtain an approximate energy spectrum.

The ground states of the effective Hamiltonian remain the ferromagnetic states, as $\hat{H}_\perp$ is zero on these states, while $\hat{H}_a$ acts to create two domain walls (up to boundary terms) and is therefore neglected.

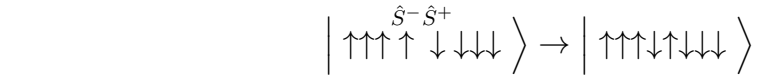


Figure 3.2: Terms in $\hat{H}_\perp$ act to create three domain walls.

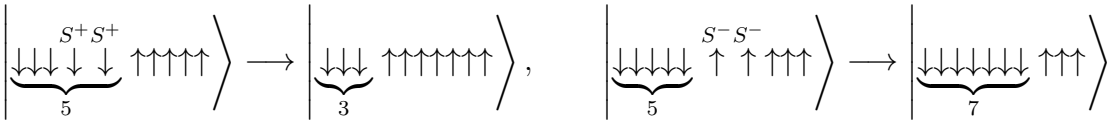


Figure 3.3: Terms in H_a act to move the domain wall by two sites in either direction.

The states with a single domain wall can be written as $|j, Q\rangle$, where $j \in \mathbb{N}$ denotes the position of the domain wall counting from the left boundary and $Q = \pm 1$ is the *charge* of the wall, denoting whether the left domain points up or down (see figure 3.1). When $\hat{H}_\perp$ acts on such a state, the only resulting nonzero term contains *three* domain walls and is discarded (see figure 3.2). For $\hat{H}_a$, the only resulting terms with at most one domain wall are $J_a(|j+2, Q\rangle + |j-2, Q\rangle)$ (see figure 3.3). Thus, the additional terms in the Hamiltonian act to move the domain wall two lattice sites in either direction. This will have the effect of giving walls a kinetic energy, as can be seen by constructing momentum superpositions of single-wall states. Choosing units so that the lattice spacing is 1, these are

$$|k, Q\rangle = \sum_j e^{ikj} |j, Q\rangle .$$

The action of the Hamiltonian on such a state is

$$\begin{aligned}
\hat{H} |k, Q\rangle &= \frac{1}{2} J_z |k, Q\rangle + J_a \sum_j e^{ikj} \left(|j+2, Q\rangle + |j-2, Q\rangle\right) \\
&= \frac{1}{2} J_z |k, Q\rangle + J_a \sum_j \left(e^{ik(j-2)} + e^{ik(j+2)}\right) |j, Q\rangle \\
&= \left(\frac{1}{2} J_z + 2 J_a \cos 2k\right) |k, Q\rangle . \qquad (3.13)
\end{aligned}$$

From this we deduce that the single domain wall states with a given charge form an energy band with dispersion relation

$$E_k = \frac{1}{2} J_z + 2 J_a \cos 2k. \qquad (3.14)$$

This is of the same form that we used to derive Bloch oscillations (section 3.1.1). If one could add a force on the domain wall a similar derivation is expected to apply, yielding Bloch oscillations in the position of the domain wall.

3.2.3 Effect of magnetic field

One can create an effective force on the domain wall by adding an external magnetic field. This magnetic field leads to a Zeeman coupling of individual spins to the field. Orienting the field along the z direction, we model it by an additional term in the Hamiltonian given by

$$\hat{H}_{\text{ext}} = -h_z \sum_{j=1}^{N} \hat{S}_j^z, \tag{3.15}$$

where $h_z = g\mu_B(B + B_{\text{int}})$ is the effective Zeeman energy. The B_{int} term is added to account for inter-chain couplings.

With this addition, states will carry an additional energy given by $-\frac{1}{2}h_z(N_\uparrow - N_\downarrow)$, where $N_{\uparrow\downarrow}$ is the number of spins oriented in the given direction. For a domain wall state with position j and charge Q, the difference is $Q(2j - N)$. This means that a domain wall state carries a potential energy linear in its position, in effect creating a constant force on the domain wall, with strength $-h_z Q$. Carrying out the same calculation as in section 3.1.1, one finds that the position of the domain wall oscillates as

$$j(t) = \text{const.} - A_B \cos \omega_B t, \tag{3.16}$$

where the Bloch frequency is simply $\hbar\omega_B = 2h_z$. The amplitude works out to be $A_B = 2J_a/h_z$.

One can also perform a completely quantum derivation of this behavior by diagonalizing the effective Hamiltonian with an additional magnetic field present from the outset. The action of the Hamiltonian on a domain wall restricted to the single-wall subspace is

$$\hat{H}\,|j, Q\rangle = \frac{1}{2}J_z\,|j, Q\rangle + J_a\left(|j+2, Q\rangle + |j-2, Q\rangle\right) - h_z Q j\,|j, Q\rangle\,, \tag{3.17}$$

where introduced a global energy shift of $\frac{1}{2}h_z N$ so that the ground-state energy is zero. The j dependence in the last term makes it nontrivial to diagonalize the system, but it can be done by parameterizing the eigenstates with Bessel functions and using their recursion relation to show that such a parameterization is diagonal in $\hat{H}$ (Kyriakidis and Loss, 1998). This results in energy states $|n\rangle$ localized near position $j = n$, with energies $E_n = \frac{1}{2}J_z + h_z n$.

This ladder of equally spaced energy levels is dubbed the Wannier-Zeeman ladder. While the energy level spacing is half the Bloch frequency we found above, the effective Hamiltonian only couples states where n differs by two. This means that we effectively have two separate ladders, each with energy spacing $\omega_B = 2h_z$.

3.2.4 Two domain walls

While restricting to a single domain wall is presumably a good approximation in the absence of a magnetic field, it seems questionable when such a field is present. This is most easily seen by returning to the simple Ising model. Adding a magnetic field, we find a dramatically different picture of the low-lying excitations.

Consider first the ferromagnetic states. The degeneracy of these states is lifted by the magnetic field, with an energy difference linear in the number of particles, which is taken to be very

$$|j,l\rangle = \left|\underbrace{\uparrow\uparrow\ldots\uparrow}_{j}\underbrace{\downarrow\downarrow\ldots\downarrow}_{l}\underbrace{\uparrow\uparrow\ldots\uparrow}_{N-j-l}\right\rangle$$

Figure 3.4: Labeling of single-domain states

large. Thus, one of the ferromagnetic states remains the ground state, while the other is highly excited.

Measuring energy relative to the ground state, the single domain wall states now have energy $\frac{1}{2}J_z + \frac{1}{2}h_zN(1-Q) + h_zQn$. This energy will be very large unless the domain wall is close to the boundary. In the above discussion we assumed such states to be low-lying excitations. However, allowing another domain wall gives rise to states with much lower energy.

The reason for this is that the two domain walls can be brought close together, leaving the number of wrongly oriented spins small. A more proper analysis should therefore work in the subspace of two domain walls. We will call states in this subspace *single-domain* states.

We can label such states by $|j,l\rangle$, with j the position of the first domain wall and l the distance between walls (see figure 3.4). The energy of such a state is $J_z + h_zl$, where the l can be as small as 1. We also include the ground state by allowing $l = 0$, in which case the j quantum number is degenerate. The action of $\hat{H}_a$ and $\hat{H}_\perp$ on states in this subspace is somewhat complex. Shinkevich and Syljuåsen (2012) diagonalized the effective Hamiltonian of this system and found energies given by $E_n = J_z + h_z\mu_n$, where μ_n is given by equations involving the ratio of two Bessel functions:

$$\frac{J_{-(\mu_n+1)/2}(x)}{J_{1-(\mu_n+1)/2}(x)} = \frac{J_\perp}{J_a}, \qquad n \text{ odd}, \tag{3.18}$$

$$\frac{J_{-\mu_n/2}(x)}{J_{1-\mu_n/2}(x)} = 0, \qquad n \text{ even},\ p \neq 0, \tag{3.19}$$

$$\frac{J_{-\mu_n/2}(x)}{J_{1-\mu_n/2}(x)} = \frac{x}{4(y+\mu_n)}, \qquad n \text{ even},\ p = 0. \tag{3.20}$$

Here $x = 2J_a\cos p/h_z$, $y = J_z/h_z$, p is the momentum of the state and $J_\nu(x)$ is the Bessel function of the first kind of order ν. One can show that the solutions of these equations satisfy $\mu_n \approx n$ for high n, but for lower n the interaction between the domain walls gives modifications of the energies. Figure 3.5 shows the excitation energy spectrum given a particular magnetic field of $h_z = 0.05J_z$.

Wavefunctions

The energy eigenstates corresponding to the above solutions are parameterized as

$$|n,p\rangle = \sum_l \psi_{nl}(p)\,|p,l\rangle\,, \tag{3.21}$$

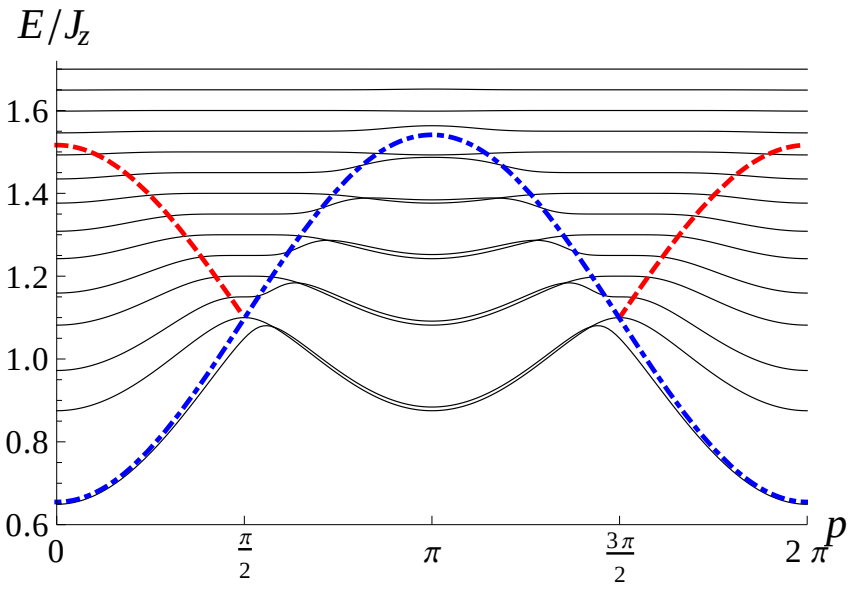


Figure 3.5: Dispersion relations for the energy bands found for $CoCl_2 \cdot 2\,H_2O$ with a magnetic field of $h_z = 0.05 J_z$. The dashed outlines indicate the boundaries between equally and non-equally spaced energy levels, as can be found by analyzing equations (3.18 – 3.20). Reproduced from Shinkevich and Syljuåsen (2012).

where $|p, l\rangle$ is the momentum eigenstate with domain length l, given by

$$|p, l\rangle = e^{-ipl/2} \sum_j e^{-ipj} \, |j, l\rangle \, . \tag{3.22}$$

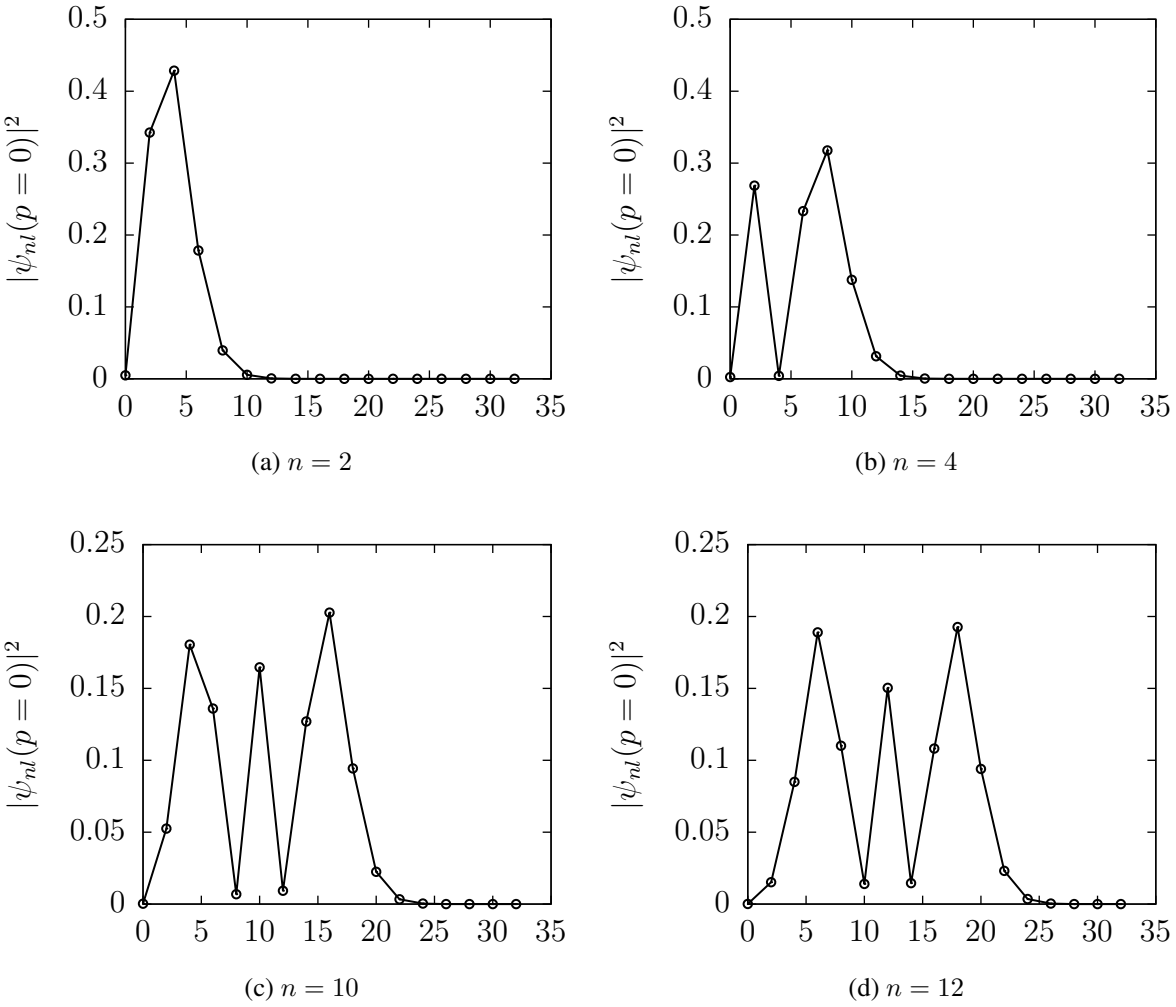
The units are such that $\hbar = 1$ and the lattice spacing is one. Given this parameterization, the expansion coefficients for odd n are

$$\psi_{nl}(p) \propto \begin{cases} J_{(l-\mu_n)/2}(x) & \text{for odd } l, \\ 0 & \text{for even } l \end{cases} . \tag{3.23}$$

The even n case needs to take the coupling to the ferromagnetic state $l = 0$ into account, giving the somewhat more complex

$$\psi_{nl}(p) \propto \begin{cases} J_{(l-\mu_n)/2}(x) & \text{for even } l > 0, \\ -\frac{J_a}{E_n(p=0)} J_{(2-\mu_n)/2}(x) & \text{for } l = 0,\, p = 0, \\ 0 & \text{otherwise.} \end{cases} \tag{3.24}$$

We have plotted the expansion coefficients at $p = 0$ for some even excitations in figure 3.6. While it is tempting to think of the $|n, p\rangle$ energy eigenstates as l-long domains with momentum p, these plots show that such an interpretation is inaccurate even for large n, as they contain even larger contributions from $l = n \pm 6$. The $n = 4$ state is close to *zero* at $l = 4$, with peaks at $l = 2,\ 8$.



(a) $n = 2$

(b) $n = 4$

(c) $n = 10$

(d) $n = 12$

Figure 3.6: Populations of domain sizes in the $|n, p = 0\rangle$ energy eigenstates with four choices of n, ignoring the zero values at odd l. At low n the populations have complex behavior, with the $n = 4$ state even having a minimum at $l = 4$. At high n we see a peak at $l = n$ flanked by higher peaks at $l = n \pm 6$. Note the different y scales between the first two and last two plots.

3.3 Rabi oscillation

As we saw in section 3.1.3, we need to populate at least two states from the energy ladder to achieve oscillations. Shinkevich and Syljuåsen (2013) proposed using lasers to excite such states, by tuning the laser so that it resonates with one of the excitation energies. This will induce oscillations between the ground state and the excited state, in a process known as *Rabi oscillation*.

The laser is modeled simply as an oscillating magnetic field, represented as an additional time-dependent term in the Hamiltonian, given by

$$\hat{H}_l(t) = V_0 \cos \omega t \sum_j \hat{S}_j^z. \tag{3.25}$$

The electric field of the laser is unimportant here, as the degenerate d-shell states of the chain have zero electric dipole moment. While the resulting Hamiltonian is suitable for numerical calculations, we would also like to gain some theoretical insight into the Rabi process. For this purpose the complete Hamiltonian is too complicated, so we consider here a simplified model of the excitation process.

We will think of our system as consisting of only two states: The ground state $|0\rangle$ and the desired excited state $|1\rangle$. In this subspace the Hamiltonian is of the form

$$\hat{H}(t) = \hat{H}_0 + \hat{V}(t), \quad \hat{H}_0 |n\rangle = E_n |n\rangle , \tag{3.26}$$

where $\hat{H}_0$ is the unperturbed system and $\hat{V}(t) = V_0 \cos \omega t \hat{O}$ is the applied magnetic field. The observable $\hat{O}$ describes how the magnetic field couples to the states in the two-dimensional subspace. This operator should couple the ground and excited states for anything to happen: We must have $\langle 0| \hat{O} |1\rangle \neq 0$.

When solving this system, it is convenient to parameterize the general state as

$$|\psi(t)\rangle = \sum_n a_n(t) e^{-ic_n(t)} |n\rangle , \tag{3.27}$$

where $c_n(t) = E_n t + \frac{V_0}{\omega} \sin \omega t \langle n| \hat{O} |n\rangle$ is an uninteresting phase variation of the coefficients due to the diagonal part of the Hamiltonian. Notice in particular that

$$\dot{c}_n(t) = E_n + V_0 \cos \omega t \langle n| \hat{O} |n\rangle = E_n + \langle n| \hat{V}(t) |n\rangle = \langle n| \hat{H} |n\rangle . \tag{3.28}$$

Inserting into the Schrödinger equation we obtain

$$i \frac{\mathrm{d}}{\mathrm{d}t} |\psi(t)\rangle = \hat{H} |\psi(t)\rangle \tag{3.29}$$

$$i \sum_n e^{-ic_n(t)} \left(\dot{a}_n(t) - i \dot{c}_n(t) a_n(t) \right) |n\rangle = \sum_n a_n(t) e^{-ic_n(t)} \hat{H} |n\rangle \tag{3.30}$$

$$i \dot{a}_m(t) + \dot{c}_m(t) a_m(t) = \sum_n a_n e^{i(c_m(t) - c_n(t))} \langle m| \hat{H} |n\rangle , \tag{3.31}$$

where we multiplied from the left with $e^{ic_m(t)} \langle m|$. Noticing that the second term on the left

hand side equals the $n = m$ term on the right hand side, this reduces to

$$i\dot{a}_m(t) = \sum_{n \neq m} a_n e^{i(c_m(t) - c_n(t))} V_{mn}(t), \tag{3.32}$$

which is equivalent to the pair of differential equations given by

$$\dot{a}_0(t) = -i a_1(t) e^{-[c_1(t) - c_0(t)]} V_0 O_{01} \cos \omega t, \tag{3.33}$$

$$\dot{a}_1(t) = -i a_0(t) e^{c_1(t) - c_0(t)} V_0 (O_{01})^* \cos \omega t. \tag{3.34}$$

In order to solve this analytically we need to make some approximations. First we will assume that we can ignore the diagonal part of the perturbation, so that $O_{11} = O_{00} = 0$, as such a perturbation is expected to have a small effect. This simplifies the exponentials to $c_1(t) - c_0(t) = E_1 t - E_0 t = ht$, were h is the energy gap. Defining $k = V_0 O_{01}$ and expanding the cosine in terms of exponentials this leaves us with

$$\dot{a}_0(t) = -\frac{i}{2} a_1(t) k \left(e^{-i(h+\omega)t} + e^{-i(h-\omega)t} \right), \tag{3.35}$$

$$\dot{a}_1(t) = -\frac{i}{2} a_0(t) k^* \left(e^{i(h+\omega t)t} + e^{i(h-\omega)t} \right). \tag{3.36}$$

where $k = V_0 O_{01}$. In order to consider the resonance behavior of the system we assume that the frequency ω is close to the energy gap h. When that is the case, the exponentials in each equation describe two oscillating terms, one of which oscillates much faster than the other. Making use of the *rotating wave approximation*, we neglect the rapidly oscillating term, as it quickly averages to zero. Given this, we are left with

$$\dot{a}_0 = -\frac{i}{2} a_1 k e^{-i\delta t}, \qquad \dot{a}_1 = -\frac{i}{2} a_0 k^* e^{i\delta t}, \tag{3.37}$$

where $\delta = h - \omega$ is the detuning. Differentiating the first equation, inserting the second and inserting an inverted first equation for a_1, we find

$$\ddot{a}_0 + i\delta \dot{a}_0 + \frac{|k|^2}{4} a_0 = 0, \tag{3.38}$$

with the general solution

$$a_0(t) = e^{-i\delta t/2} \left(A e^{\frac{i}{2}\omega_R t} + B e^{-\frac{i}{2}\omega_R t} \right), \tag{3.39}$$

where the Rabi frequency is given by $\omega_R = \sqrt{\delta^2 + |k|^2} = \sqrt{\delta^2 + V_0^2 |O_{01}|^2}$. For the other coefficient we find

$$a_1 = \frac{2i}{k} e^{i\delta t} \dot{a}_0 = -\frac{1}{k} e^{i\delta t/2} \left[A(\omega_R - \delta) e^{\frac{i}{2}\omega_R t} - B(\omega_R + \delta) e^{-\frac{i}{2}\omega_R t} \right]. \tag{3.40}$$

Starting in the ground state we have $a_0(0) = 1, a_1(0) = 0$. From this we find $A + B = 1$ and

$$A(\omega_R - \delta) - B(\omega_R + \delta) = 0, \tag{3.41}$$

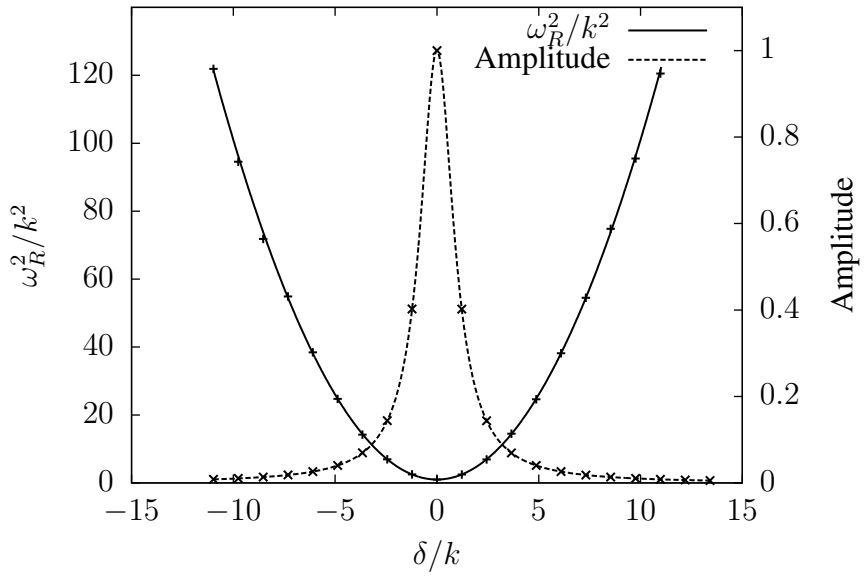


Figure 3.7: Dependence of Rabi frequency and amplitude from the numerical solution of equations (3.33 - 3.34) (points), compared with the approximate analytical solution (solid curves).

with the solutions $A = \frac{1}{2}(1 + \delta/\omega_R), B = \frac{1}{2}(1 - \delta/\omega_R)$. This yields for our state,

$$a_0(t) = e^{-i\delta t/2}\left(\cos\frac{1}{2}\omega_R t + i\frac{\delta}{\omega_R}\sin\frac{1}{2}\omega_R t\right) \tag{3.42}$$

$$a_1(t) = \frac{ik^*}{\omega_R}e^{i\delta t/2}\sin\frac{1}{2}\omega_R t. \tag{3.43}$$

The occupancy of the excited state is

$$|a_1(t)|^2 = \frac{|k|^2}{\omega_R^2}\sin^2\frac{1}{2}\omega_R t = \frac{|k|^2}{2\omega_R^2}(1 - \cos\omega_R t). \tag{3.44}$$

This gives us the following picture for the time-evolution: At resonance, there is a simple oscillation between the ground and excited states with frequency $\omega_R = |O_{01}|V_0$ given by the coupling strength of the perturbation between the states. When detuning is added the frequency increases, while the maximum occupancy of the excited state is reduced to

$$|a_1|^2_{\text{max}} = \frac{|k|^2}{\omega_R^2} = \frac{V_0^2|O_{01}|^2}{\delta^2 + V_0^2|O_{01}|^2}. \tag{3.45}$$

When the detuning is smaller than $V_0|O_{01}|$ this is close to one, so we have an approximately resonant oscillation. With larger detuning the amplitude decreases while the frequency increases. When looking for such a resonance, we should therefore aim to determine the frequency to within the coupling value $V_0|O_{01}|$.

Experimentally, the oscillating population will show up as an oscillation in the magnetization given by $N_\downarrow = (N_\downarrow^1 - N_\downarrow^0)|a_1|^2$, where l_n is the magnetization of the $|n\rangle$ state.

In order to verify the validity of the approximations used in this section, we solved equations

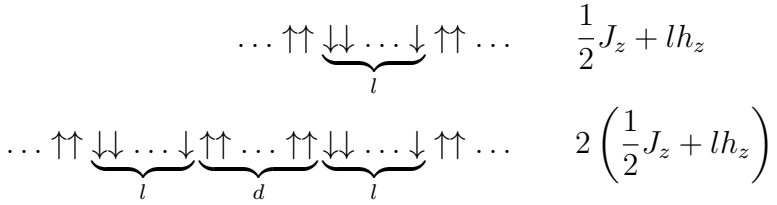


Figure 3.8: A single domain in the Ising model has energy depending on the domain size. A state of two similar domains separated by distance d has twice that energy.

(3.33 – 3.34) numerically using some typical parameters for our problem. The result is shown in figure 3.7, which shows excellent agreement with the analytical solution.

3.3.1 Several excitations

In the above discussion we assumed that the laser-excitation process is accurately described as a two-level resonance problem. There are two ways this can fail to be the case. Firstly, if several excitations have energies that lie closely together compared to the strength of the coupling $V_0|O_{01}|$, we will obtain Rabi oscillations to several different states simultaneously. This is known as off-resonant tunneling and happens when the laser amplitude is big.

In this section will focus on another way the system can fail to act as a two-level system, namely when there is an excited state $|2\rangle$ with energy $E_2 \approx E_1 + h$ at about the same energy difference from $|1\rangle$ as the energy gap $h = E_1 - E_0$. With such an energy spectrum the laser will induce Rabi oscillations between $|1\rangle$ and $|2\rangle$ in addition to the oscillation between $|0\rangle$ and $|1\rangle$, so that we need to include all three states for a correct description.

Such energy spectra typically occur when two similar excitations can be packed into the same state. In the system we are considering, this occurs when we create two domains of the same size (see figure 3.8). In the Ising model approximation, states with two domains have exactly twice the energy of a single-domain state. When the non-Ising terms are added, we should expect there to be an additional interaction energy between the domains, depending on the distance d between the domains.

In order to model this situation we write $|l\rangle$ for the left-hand domain state, $|r\rangle$ for the right-hand domain state and $|lr\rangle$ for the state containing both excitations. Since our system including the perturbation is translationally invariant, we expect only zero-momentum states to be relevant, so we write $|1'\rangle = \frac{1}{\sqrt{2}}(|l\rangle + |r\rangle)$ and $|2'\rangle = |lr\rangle$, adding primes to distinguish the states from the excited state we considered above. The remaining possibility, $|l\rangle - |r\rangle$ with momentum π, is then ignored.

We also assume that the $|l\rangle$ and $|r\rangle$ states couple similarly to the perturbation as the single excited state did before, so that $\langle l|\,\hat{O}\,|0\rangle = \langle r|\,\hat{O}\,|0\rangle = O_{10}$, where O_{10} is as above. With this assumption, the coupling to the state $|1'\rangle$ is

$$\langle 1'|\,\hat{O}\,|0\rangle = \frac{1}{\sqrt{2}}\Big(\langle l|\,\hat{O}\,|0\rangle + \langle r|\,\hat{O}\,|0\rangle\Big) = \frac{2}{\sqrt{2}}O_{10} = \sqrt{2}O_{10}, \tag{3.46}$$

meaning that the coupling increases by $\sqrt{2}$. In a similar manner, we assume that creating a

second domain in a single-domain state is a similar process to creating a single-domain state from the ground state. This means that $\langle lr|\,\hat{O}\,|l\rangle = \langle lr|\,\hat{O}\,|r\rangle = O_{10}$, so that $\langle 2'|\,\hat{O}\,|1'\rangle = \sqrt{2}O_{10}$.

If we ignore the diagonal parts of the perturbation from the outset and parameterize the state as $|\psi(t)\rangle = \sum_n e^{-i(E_n-E_0)t}a_n\,|n'\rangle$, we find a system of differential equations given by

$$i\dot{a}_m(t) = \sum_{n\neq m} a_n e^{i(E_m-E_n)t}V_0\cos\omega t\,\langle m'|\,\hat{O}\,|n'\rangle\,. \tag{3.47}$$

For the energies we have $E_r - E_0 = E_l - E_0 = h$, so that $E_1 - E_0 = h$. The state with two domains has energy at least $E_0 + 2h + \Delta$, where Δ is the interaction energy. Assuming resonance and using the rotating wave approximation we are left with

$$\dot{a}_0 = -\frac{i}{\sqrt{2}}a_1 k, \tag{3.48}$$

$$\dot{a}_1 = -\frac{i}{\sqrt{2}}a_0 k^* - \frac{i}{\sqrt{2}}a_2 k e^{-i\Delta t}, \tag{3.49}$$

$$\dot{a}_2 = -\frac{i}{\sqrt{2}}a_1 k^* e^{i\Delta t}, \tag{3.50}$$

where $k = V_0O_{01}$ as before. Manipulating these equations gives a third order equation for a_0:

$$\dddot{a}_0 + i\Delta\ddot{a}_0 + |k|^2\dot{a}_0 + \frac{i}{2}\Delta|k|^2 a_0 = 0. \tag{3.51}$$

This equation can be solved by finding the roots of the cubic characteristic polynomial. However, as such roots get very messy we will rather consider some limiting behaviors to gain insight into the system.

Consider first $\Delta = 0$, meaning that packing two domains into a state carries no energy penalty. Then the differential equation reduces to a simple harmonic equation for $\dot{a}_0 \propto a_1$. Starting in the ground state, we find the solution

$$a_0(t) = \frac{1}{2} + \frac{1}{2}\cos\tfrac{1}{2}wt, \quad a_1(t) = -\frac{i}{\sqrt{2}}\sin\tfrac{1}{2}wt, \quad a_2(t) = -\frac{1}{2} + \frac{1}{2}\cos\tfrac{1}{2}wt, \tag{3.52}$$

where $w = 2|k|$. It therefore seems like the Rabi frequency has doubled compared to the two-level system. However, if we calculate the magnetization, we find (assuming the $|2\rangle$ state has magnetization $2l$),

$$\begin{aligned} N_\downarrow(t) &= l|a_1(t)|^2 + 2l|a_2(t)|^2 = \frac{l}{2}\sin^2\tfrac{1}{2}wt + \frac{l}{2} + \frac{l}{2}\cos^2\tfrac{1}{2}wt - \frac{l}{2}\cos\tfrac{1}{2}wt \\ &= \frac{l}{2}\left(1 - \cos\tfrac{1}{2}wt\right) = \frac{l}{2}\left(1 - \cos\omega_R t\right), \end{aligned} \tag{3.53}$$

so the magnetization still oscillates with frequency $\omega_R = |k|$, despite the different behavior of the populations.

Next we consider $\Delta \to \infty$, meaning that the interaction energy is much larger than the energy spacings. In this limit, the terms proportional to Δ dominate over the other terms,

leaving the harmonic equation

$$\ddot{a}_0 + \frac{1}{2}|k|^2 a_0 = 0. \tag{3.54}$$

Starting in the ground state, the large energy of the $|2'\rangle$ state ensures that this state will not be populated at all. This leaves us with simple harmonic oscillation between the ground state and the $|1'\rangle$ state, so that the magnetization oscillates with frequency $\omega_B = \sqrt{2}|k| = \sqrt{2}\omega_R$. The effect of strong interactions between domains is therefore to increase the oscillation frequency of the magnetization. We call this effect *blocking*.

For intermediate interaction energies we have to consider the complete third-order differential equation, giving more complex behavior. One can show that the oscillation splits into two components, one of which goes to zero for high interaction, while the other ends up at the frequency $\omega_B = \sqrt{2}\omega_R$.

Informed by this analysis, one can expect similar effects to happen when one can fit n domains into the state. If the domains do not interact there will be no effect on the oscillation frequency of the magnetization. With strong interaction the blocking effect will lead to a $\sqrt{n}$ increase in the frequency, while for intermediate interactions more complex behavior can occur.

Chapter 4

Time-evolving block decimation

This chapter introduces the simulation method we will use for simulating many-particle quantum systems, namely *time-evolving block decimation*, or TEBD in short. The method is motivated by quantum information theoretical considerations of a specific class of systems thought to encompass many realistic, one-dimensional systems.

The general problem we will be interested in is that of a collection of N particles, each described by the same d-dimensional Hilbert space $\mathcal{H}$. As we have seen, in order to consider the complete system we have to perform calculations in $\mathcal{H}^{\otimes N}$, with d^N dimensions. The most straightforward way of calculating the properties of such a system is to diagonalize the $d^N \times d^N$ matrix representing the Hamiltonian, yielding the energy spectrum. Then one can use the time-development operator introduced in section 2.4 to calculate the time-dependence of the system.

The obvious problem with this approach is the size of the representations: States are d^N-dimensional vectors, while operators are represented with $d^N \times d^N$ matrices, so that any explicit calculations get very quickly out of hand. This is the central problem in many-body quantum mechanics. Partial solutions range from quantum monte carlo, where one uses random sampling to avoid considering the entire space, to the variational method, where one restricts Hilbert space to a hopefully representative, low-dimensional subspace.

TEBD can be viewed as a variational method in the class of *matrix product states* (MPS), which is a class of states for one-dimensional systems. However, unlike many other variational approaches, this class is large enough to describe the complete low-dimensional spectrum of many Hamiltonians. This enables simulations of time-evolution as well as approximating the ground state. In this thesis, however, we will motivate the method without using variational language.

The crucial fact that TEBD exploits is that Hamiltonians with only *local interactions* tend to have a low-energy spectrum with a small amount of entanglement, as long as there is an energy gap. This has been proved for one-dimensional gapped Hamiltonians in the context of matrix product states by Verstraete and Cirac (2006). One can get an intuitive sense for why it should be the case by first considering a system of noninteracting particles. The Hamiltonian for such a system separates into commuting parts describing each particle. This lets us find the energy spectrum for each particle separately, from which we can generate the energy spectrum of the entire system by tensoring single-particle states together. Such product are not entangled, so the energy spectrum of a noninteracting system shows no entanglement.

The smallest possible change one can make to the noninteracting system is to add nearest-

neighbor interactions. With this addition the above procedure breaks down, but since the system still looks somewhat like the noninteracting one, one might expect that the amount of entanglement stays small.

4.1 Representation

In order to bring out the entanglement properties of the state and make the necessary approximations, we make use of Schmidt decompositions. With open boundary conditions, one can make a cut between any pair of particles to divide the system in two. This means that there is a different Schmidt decomposition for each pair of particles. Since all of these decompositions will be useful to us, we devise a representation of the state which encodes all the decompositions simultaneously, as first described in Vidal (2003).

Note that the manipulations in this section serve only to theoretically motivate the expansion we will end up with; it is in general not feasible to perform them directly. See below for how to actually obtain the state of interest.

The starting point is an expansion of our state $|\psi\rangle$ in the product basis:

$$|\psi\rangle = \sum_{i_1,i_2\ldots i_N} c^{[1]}_{i_1 i_2\ldots i_N} |i_1\rangle |i_2\rangle \ldots |i_N\rangle , \tag{4.1}$$

where the superscript index on the expansion coefficients will help us to keep track of how far we have come in our construction. Next we will build the representation inductively by a series of Schmidt decompositions. We begin by separating out the first particle from the rest of the system. This requires a singular value decomposition of the matrix M given by $M_{i_1,(i_2 i_3\ldots i_N)} = c^{[1]}_{i_1 i_2\ldots i_N}$, which results in new matrices given by $M = L\Sigma R^\dagger$. Here L and R are unitary and Σ is diagonal with Schmidt coefficients σ_α on the diagonal. In coordinate form we have

$$c^{[1]}_{i_1 i_2\ldots i_N} = \sum_\alpha L_{i_1,\alpha}\sigma_\alpha R^*_{(i_2 i_3\ldots i_N),\alpha}. \tag{4.2}$$

Setting $\Gamma^{[1]i_1}_\alpha = L_{i_1,\alpha}$, $\lambda^{[1]}_\alpha = \sigma_\alpha$ and $c^{[2]\alpha}_{i_2 i_3\ldots i_N} = R^*_{(i_2 i_3\ldots i_N),\alpha}$ we obtain the Schmidt decomposition

$$|\psi\rangle = \sum_{\alpha_1} \lambda^{[1]}_{\alpha_1} \left|\Phi^{[1]}_{\alpha_1}\right\rangle \left|\Phi^{[2\ldots N]}_{\alpha_1}\right\rangle , \tag{4.3}$$

where

$$\left|\Phi^{[1]}_{\alpha_1}\right\rangle = \sum_{i_1} \Gamma^{[1]i_1}_{\alpha_1} |i_1\rangle , \tag{4.4}$$

$$\left|\Phi^{[2\ldots N]}_{\alpha_1}\right\rangle = \sum_{i_2 i_3\ldots i_N} c^{[2]\alpha_1}_{i_2 i_3\ldots i_N} |i_2 i_3 \ldots i_N\rangle . \tag{4.5}$$

These Schmidt vectors are orthonormal due to the unitarity of L and R.

Now we want to separate out the first *two* particles from the rest of the system. Writing the

state as

$$|\psi\rangle = \sum_{\alpha_1 i_2} \sum_{i_3 \ldots i_N} \lambda^{[1]}_{\alpha_1} c^{[2]\alpha_1}_{i_2 i_3 \ldots i_N} \left(\left|\Phi^{[1]}_{\alpha_1}\right\rangle |i_2\rangle \right) |i_3 \ldots i_N\rangle \,, \tag{4.6}$$

we see that we require a singular value decomposition of the matrix given by $M_{(i_2\alpha_1),(i_3\ldots i_N)} = \lambda^{[1]}_{\alpha_1} c^{[2]\alpha_1}_{i_2 i_3 \ldots i_N}$. With $M = L\Sigma R^\dagger$ we obtain

$$\lambda^{[1]}_{\alpha_1} c^{[2]\alpha_1}_{i_2 i_3 \ldots i_N} = \sum_{\alpha_2} L_{(i_2\alpha_1),\alpha_2} \sigma_{\alpha_2} R^*_{(i_3 \ldots i_N),\alpha_2}. \tag{4.7}$$

We now make the definitions

$$\Gamma^{[2]i_2}_{\alpha_1\alpha_2} = \frac{1}{\lambda^{[1]}_{\alpha_1}} L_{(i_2\alpha_1),\alpha_2}, \quad \lambda^{[2]}_{\alpha_2} = \sigma_{\alpha_2}, \quad c^{[3]\alpha_2}_{i_3 \ldots i_N} = R^*_{(i_3 \ldots i_N),\alpha_2}. \tag{4.8}$$

With these we can define the Schmidt vectors

$$\left|\Phi^{[1,2]}_{\alpha_2}\right\rangle = \sum_{i_2\alpha_1} \lambda^{[1]}_{\alpha_1} \Gamma^{[2]i_2}_{\alpha_1\alpha_2} \left|\Phi^{[1]}_{\alpha_1}\right\rangle |i_2\rangle \,, \tag{4.9}$$

$$\left|\Phi^{[3\ldots N]}_{\alpha_2}\right\rangle = \sum_{i_3 \ldots i_N} c^{[3]\alpha_2}_{i_3 \ldots i_N} |i_3 \ldots i_N\rangle \,, \tag{4.10}$$

which yields the Schmidt decomposition:

$$\begin{aligned} |\psi\rangle &= \sum_{\alpha_2} \lambda^{[2]}_{\alpha_2} \left|\Phi^{[1,2]}_{\alpha_2}\right\rangle \left|\Phi^{[3\ldots N]}_{\alpha_2}\right\rangle \\ &= \sum_{\alpha_1\alpha_2} \sum_{i_1 i_2} \Gamma^{[1]i_1}_{\alpha_1} \lambda^{[1]}_{\alpha_1} \Gamma^{[2]i_2}_{\alpha_1\alpha_2} |i_1 i_2\rangle \left|\Phi^{[3\ldots N]}_{\alpha_2}\right\rangle . \end{aligned} \tag{4.11}$$

Note that the definitions of the new left-hand Schmidt vectors $\left|\Phi^{[1,2]}_{\alpha_2}\right\rangle$ are given in terms of the previous Schmidt vectors $\left|\Phi^{[1]}_{\alpha_1}\right\rangle$. An analogous property can be shown for the right-hand set of new Schmidt vectors:

$$\begin{aligned} \left|\Phi^{[2\ldots N]}_{\alpha_1}\right\rangle &= \sum_{i_2 i_3 \ldots i_N} c^{[2]\alpha_1}_{i_2 i_3 \ldots i_N} |i_2 i_3 \ldots i_N\rangle \\ &= \sum_{i_2\alpha_2} \sum_{i_3 \ldots i_N} \Gamma^{[2]i_2}_{\alpha_1\alpha_2} \lambda^{[2]}_{\alpha_2} c^{[3]\alpha_2}_{i_3 \ldots i_N} |i_2\rangle |i_3 \ldots i_N\rangle \\ &= \sum_{i_2\alpha_2} \Gamma^{[2]i_2}_{\alpha_1\alpha_2} \lambda^{[2]}_{\alpha_2} \left|\Phi^{[3\ldots N]}_{\alpha_2}\right\rangle . \end{aligned} \tag{4.12}$$

These properties are part of what makes it easy to move between different Schmidt decomposition.

Iterating the above procedure we obtain the expansion

$$|\psi\rangle = \sum_{\alpha_1 \ldots \alpha_{N-1}} \sum_{i_1 \ldots i_N} \Gamma^{[1]i_1}_{\alpha_1} \lambda^{[1]}_{\alpha_1} \Gamma^{[2]i_2}_{\alpha_1\alpha_2} \ldots \Gamma^{[N-1]i_{N-1}}_{\alpha_{N-2}\alpha_{N-1}} \lambda^{[N-1]}_{\alpha_{N-1}} \Gamma^{[N]i_N}_{\alpha_{N-1}} |i_1 \ldots i_N\rangle \,. \tag{4.13}$$

This representation allows us to write the Schmidt decomposition between particles n and $n+1$

as

$$|\psi\rangle = \sum_{\beta} \lambda_{\beta}^{[n]} \left|\Phi_{\beta}^{[1\ldots n]}\right\rangle \left|\Phi_{\beta}^{[n+1\ldots N]}\right\rangle . \tag{4.14}$$

where the Schmidt vectors can be computed from Schmidt vectors for smaller subsystems as

$$\left|\Phi_{\beta}^{[1\ldots n]}\right\rangle = \sum_{\alpha i} \lambda_{\alpha}^{[n-1]} \Gamma_{\alpha\beta}^{[n]i} \left|\Phi^{[1\ldots n-1]}\right\rangle |i\rangle , \tag{4.15}$$

$$\left|\Phi_{\beta}^{[n+1\ldots N]}\right\rangle = \sum_{\gamma j} \Gamma_{\beta\gamma}^{[n+1]j} \lambda_{\gamma}^{[n+1]} |j\rangle \left|\Phi_{\gamma}^{[n+2\ldots N]}\right\rangle . \tag{4.16}$$

An example of the utility of the representation is that we can separate any neighboring pair of particles from the rest of the system, with

$$|\psi\rangle = \sum_{ij} \sum_{\alpha\beta\gamma} \lambda_{\alpha}^{[n-1]} \Gamma_{\alpha\beta}^{[n]i} \lambda_{\beta}^{[n]} \Gamma_{\beta\gamma}^{[n+1]j} \lambda_{\gamma}^{[n+1]} \left|\Phi_{\alpha}^{[1\ldots n-1]}\right\rangle \left|ij\right\rangle \left|\Phi_{\gamma}^{[n+2\ldots N]}\right\rangle . \tag{4.17}$$

4.1.1 Product states

A simple special case of the above representation is for product states. Consider a general product state $|\psi\rangle = |\psi_1\rangle |\psi_2\rangle \ldots |\psi_N\rangle$, expanded as in equation (4.13). Since this is a product state all the Schmidt ranks are one, which by normalization forces the Schmidt coefficients to equal one. Thus the representation simplifies to

$$\begin{aligned} |\psi\rangle &= \sum_{i_1 i_2 \ldots i_N} \Gamma_{1}^{[1]i_1} \Gamma_{11}^{[2]i_2} \ldots \Gamma_{1}^{[N]i_N} |i_1 i_2 \ldots i_N\rangle \\ &= \left(\sum_{i_1} \Gamma^{[1]i_1} |i_1\rangle\right) \otimes \left(\sum_{i_2} \Gamma^{[2]i_2} |i_2\rangle\right) \otimes \ldots \otimes \left(\sum_{i_N} \Gamma^{[N]i_N} |i_N\rangle\right) , \end{aligned} \tag{4.18}$$

where we dropped the lower indices on the Γ coefficients, which are now superfluous. This equation needs to hold separately at every particle, i.e. $|\psi_n\rangle = \sum_i \Gamma^{[n]i} |i\rangle$ for all n.

The result is a simple way to construct the TEBD representation for a product state: Set all the Schmidt ranks equal to one, set all the Schmidt coefficients equal to one, and finally set all $\Gamma^{[n]i} = \langle i|\psi_n\rangle$.

4.2 Truncating the matrices

So far everything we have done has been exact, so we should expect the decomposition we have obtained to have the same size problems as we had before. Indeed, when decomposing a general state $|\psi\rangle$ as

$$|\psi\rangle = \sum_{\beta} \lambda_{\beta}^{[n]} \left|\Phi_{\beta}^{[1\ldots n]}\right\rangle \left|\Phi_{\beta}^{[n+1\ldots N]}\right\rangle , \tag{4.19}$$

the Schmidt rank χ will in general be close to $\min\left\{d^n, d^{N-n-1}\right\}$. However, by the argument given in section 2.6, most of the Schmidt coefficients must be small in order to have a small entanglement between the subsystems. This motivates the crucial approximation in this method:

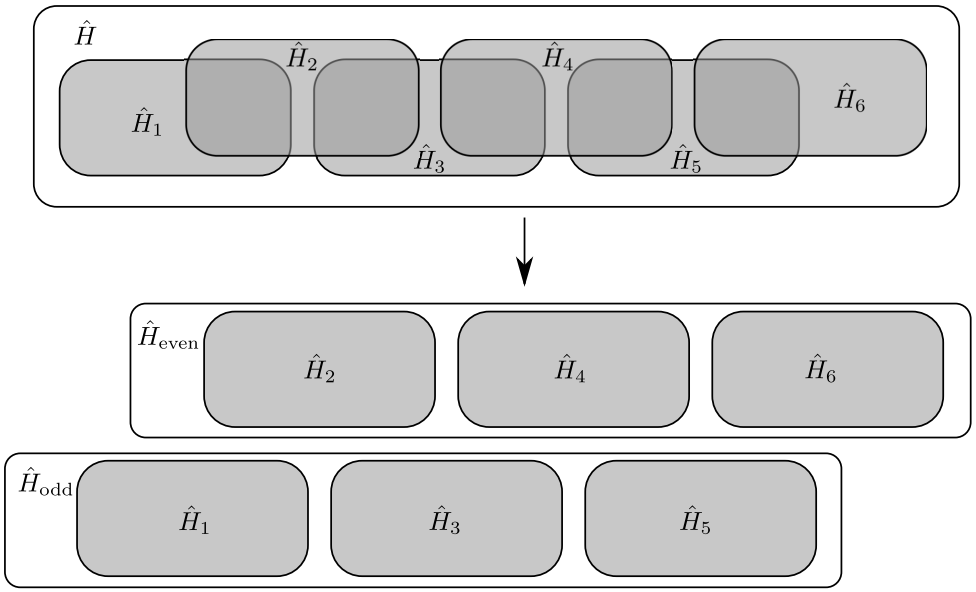


Figure 4.1: Illustration of how the overlapping terms are split into two groups of non-overlapping terms.

We fix some maximal Schmidt rank χ_{max}. Whenever a decomposition such as (4.19) has a rank exceeding χ_{max}, we set the smallest Schmidt coefficients to zero until the rank is χ_{max}. Assuming $\lambda^{[n]}$ to be sorted in descending order, this leads to the expansion

$$|\psi\rangle = \sum_{ij} \sum_{\alpha\gamma} \sum_{\beta}^{\chi_{\text{max}}} \lambda_{\alpha}^{[n-1]} \Gamma_{\alpha\beta}^{[n]i} \lambda_{\beta}^{[n]} \Gamma_{\beta\gamma}^{[n+1]j} \lambda_{\gamma}^{[n+1]} \left|\Phi_{\alpha}^{[1\ldots n-1]}\right\rangle \left|ij\right\rangle \left|\Phi_{\gamma}^{[n+2\ldots N]}\right\rangle, \tag{4.20}$$

allowing us to drop the columns of $\Gamma^{[n]i}$ and the rows of $\Gamma^{[n+1]j}$ corresponding to the smallest Schmidt coefficients $\lambda^{[n]}$. Repeating this truncation for all pairs of particles yields a representation where all the matrices have a size of at most χ_{max}. Typically, χ_{max} will need to be only a few hundred, so the truncated representation is entirely manageable on a computer.

4.3 Decomposing time-development

In order to consider the time-development of our system, we consider short enough time-steps that the Hamiltonian is constant on each step. The time-development operator for a time-step δt is

$$\hat{U}(t, \delta t) = e^{-i\hat{H}(t)\delta t}, \tag{4.21}$$

where we choose the units such that $\hbar = 1$. As we noted in section 2.7, computing this explicitly is not practical. In order to derive a workable time-development, we will make use of the fact that the relevant information of the Hamiltonian can be represented in an efficient way.

Recall our fundamental assumption that the Hamiltonian consists only of one- and two-particle terms. Any single-particle term can be turned into two-particle terms by tensoring it with the identity operator for a neighboring system. Then we can write our Hamiltonian as $\hat{H}(t) = \sum_{n=1}^{N-1} \hat{H}_n(t)$, where $\hat{H}_n$ is a two-particle operator acting on particles n and $n+1$. The

next step is to collect these terms into two groups such that all the terms each group commute with each other.

Figure 4.1 illustrates the problem: The individual terms in the Hamiltonian act on overlapping subspaces of our system, so that they do not commute. However, if we consider only the terms with even n there is no overlap, and likewise the terms with odd n do not overlap. This motivates us to define

$$\hat{H}(t) = \hat{H}_{\text{even}}(t) + \hat{H}_{\text{odd}}(t) = \sum_{n \text{ even}} \hat{H}_n(t) + \sum_{n \text{ odd}} \hat{H}_n(t). \tag{4.22}$$

The time-development operator is then given by $\hat{U}(t, \delta t) = e^{-i\hat{H}_{\text{even}}(t)\delta t - i\hat{H}_{\text{odd}}(t)\delta t}$. Applying the Suzuki-Trotter decomposition from section 2.7, we are left with a product of time-development operators for $\hat{H}_{\text{even}}$ and $\hat{H}_{\text{odd}}$, valid for small δt. Since the terms of $\hat{H}_{\text{even}}$ commute, we can separate its exponential into smaller factors without approximation, by

$$e^{-i\hat{H}_{\text{even}}(t)\delta t} = \prod_{n \text{ even}} e^{-i\hat{H}_n(t)\delta t}, \tag{4.23}$$

and similarly for $\hat{H}_{\text{odd}}$. Each of these exponentials involve only $d^2 \times d^2$ hermitean matrices and can therefore be calculated easily.

Thus we have arrived at the following representation of the time-development operator. The complete operator is given by a Suzuki-Trotter decomposition into operators for even and odd subspaces, with $\hat{U}(t, \delta t) = f_p\left[\hat{U}_{\text{even}}(t, \delta t), \hat{U}_{\text{odd}}(t, \delta t)\right]$. The time-development operator for each subspace is then given by products of exponentials, i.e. $\hat{U}_{\text{even}}(t, \delta t) = \prod_{n \text{ even}} \exp(-i\hat{H}_n(t)\delta t)$. The individual factors of this product are small $d^2 \times d^2$ matrices which can be represented efficiently, and the product can be applied by applying all the factors in succession.

4.4 Applying operators

In the previous section we saw that developing the state $|\psi(t)\rangle$ in time amounts to the repeated application of operators each acting on two neighboring particles. Since we are working in a particular representation of the state, the action of such operators is nontrivial to compute. In this section we work out how the updated state is represented.

Consider therefore a two-particle operator $\hat{U}$ acting on particle n and $n+1$. Separating out the two particles as in (4.17) and applying $\hat{U}$, we find

$$\left|\overline{\psi}\right\rangle = \hat{U}\left|\psi\right\rangle = \sum_{ij} \sum_{\alpha\beta\gamma} \lambda_\alpha^{[n-1]} \Gamma_{\alpha\beta}^{[n]i} \lambda_\beta^{[n]} \Gamma_{\beta\gamma}^{[n+1]j} \lambda_\gamma^{[n+1]} \left|\Phi_\alpha^{[1\ldots n-1]}\right\rangle \left(\hat{U}\left|ij\right\rangle\right) \left|\Phi_\gamma^{[n+2\ldots N]}\right\rangle. \tag{4.24}$$

To use the matrix representation of $\hat{U}$ we insert the identity operator $\sum_{kl} |kl\rangle\langle kl|$ and define

$U^{ij}_{kl} = \langle ij| \hat{U} |kl\rangle$ to obtain

$$\begin{aligned}\left|\overline{\psi}\right\rangle = \hat{U}\left|\psi\right\rangle &= \sum_{ijkl}\sum_{\alpha\beta\gamma} U^{ij}_{kl}\lambda^{[n-1]}_{\alpha}\Gamma^{[n]k}_{\alpha\beta}\lambda^{[n]}_{\beta}\Gamma^{[n+1]l}_{\beta\gamma}\lambda^{[n+1]}_{\gamma}\left|\Phi^{[1\ldots n-1]}_{\alpha}\right\rangle|ij\rangle\left|\Phi^{[n+2\ldots N]}_{\gamma}\right\rangle \\ &= \sum_{ij}\sum_{\alpha\gamma}\Theta^{ij}_{\alpha\gamma}\left(\left|\Phi^{[1\ldots n-1]}_{\alpha}\right\rangle|i\rangle\right)\otimes\left(|j\rangle\left|\Phi^{[n+2\ldots N]}_{\gamma}\right\rangle\right), \end{aligned} \tag{4.25}$$

where we swapped the i, j and k, l indices. While this is a correct expression for the time-developed state, it is unclear how one can update the Γs and λs of our representation to make the above hold. We need an updated Schmidt decomposition so that

$$\left|\overline{\psi}\right\rangle = \sum_{\beta}\overline{\lambda}^{[n]}_{\beta}\left|\overline{\Phi}^{[1\ldots n]}_{\beta}\right\rangle\left|\overline{\Phi}^{[n+1\ldots N]}_{\beta}\right\rangle. \tag{4.26}$$

Such a decomposition can be found by regarding the Θ coefficients as an expansion matrix $\Theta_{(i\alpha),(j\gamma)} = \Theta^{ij}_{\alpha\gamma}$ between the two subsystems $[1\ldots n]$ and $[n+1\ldots N]$, as the parentheses in equation (4.25) indicate. Due to the truncation we performed on the previous Schmidt decomposition, the size of this matrix is at most $d\chi_{\text{max}}$, so we can perform a singular value decomposition $\Theta = L\Sigma R^{\dagger}$. Element-wise, this is

$$\Theta^{ij}_{\alpha\gamma} = \sum_{\beta} L_{(i\alpha),\beta}\sigma_{\beta}R^{*}_{(j\gamma),\beta}. \tag{4.27}$$

From this we define updated Γ and λ coefficients by

$$\overline{\Gamma}^{[n]i}_{\alpha\beta} = \frac{1}{\lambda^{[n-1]}_{\alpha}}L_{(i\alpha),\beta}, \quad \overline{\Gamma}^{[n+1]j}_{\beta\gamma} = \frac{1}{\lambda^{[n+1]}_{\gamma}}R^{*}_{(j\gamma),\beta}, \quad \overline{\lambda}^{[n]}_{\beta} = \sigma_{\beta}, \tag{4.28}$$

which leads to updated Schmidt vectors given by

$$\left|\overline{\Phi}^{[1\ldots n]}_{\beta}\right\rangle = \sum_{i\alpha}\lambda^{[n-1]}_{\alpha}\overline{\Gamma}^{[n]}_{\alpha\beta}\left|\Phi^{[1\ldots n-1]}_{\alpha}\right\rangle|i\rangle, \tag{4.29}$$

$$\left|\overline{\Phi}^{[n+1\ldots N]}_{\beta}\right\rangle = \sum_{j\gamma}\overline{\Gamma}^{[n+1]}_{\beta\gamma}\lambda^{[n+1]}_{\gamma}|j\rangle\left|\Phi^{[n+1\ldots N]}_{\gamma}\right\rangle. \tag{4.30}$$

Inserting into the updated Schmidt decomposition then gives

$$\left|\overline{\psi}\right\rangle = \sum_{ij}\sum_{\alpha\beta\gamma}\lambda^{[n-1]}_{\alpha}\overline{\Gamma}^{[n]i}_{\alpha\beta}\overline{\lambda}^{[n]}_{\beta}\overline{\Gamma}^{[n+1]j}_{\beta\gamma}\lambda^{[n+1]}_{\gamma}\left|\Phi^{[1\ldots n-1]}_{\alpha}\right\rangle\left|ij\right\rangle\left|\Phi^{[n+2\ldots N]}_{\gamma}\right\rangle, \tag{4.31}$$

which amounts to replacing the $\Gamma^{[n]i}$ and $\Gamma^{[n+1]j}$ matrices and the $\lambda^{[n]}$ vector with the updated versions.

4.4.1 Truncation again

As we noted above, the expected size of the Θ matrix is $d\chi_{\text{max}} \times d\chi_{\text{max}}$. Therefore, when performing a singular value decomposition, we should expect the size of the L, R and Σ matrices to also have this size. Tracing this through the definition of the updated Γ matrices, we see that

$$\big(\Gamma^{[1]i_1}\overset{\hat{U}_1}{\Lambda^{[1]}}\Gamma^{[2]i_2}\big)\Lambda^{[2]}\big(\Gamma^{[3]i_3}\overset{\hat{U}_3}{\Lambda^{[3]}}\Gamma^{[4]i_4}\big)\Lambda^{[4]}\big(\Gamma^{[5]i_5}\overset{\hat{U}_5}{\Lambda^{[5]}}\Gamma^{[6]i_6}\big)\Lambda^{[6]}\ldots$$

Figure 4.2: Illustration of the independence of updating particle pairs. Applying the operators only modifies the coefficients inside the parentheses, while also reading the lambda coefficients on either side.

the size of $\lambda^{[n]}$ has grown by a factor of d. If we let this happen every time step, our matrices will have grown by a factor of d^n after n steps. In order to avoid an exponential growth of the matrices with time, we will therefore need to truncate the updated matrices back to χ_{max}, in the same way as described in section 4.2.

4.4.2 Independence of updates

There is a nice property about the above update procedure which we will make use of later. We have seen that part of the process for performing a single time step requires us to apply two-particle operators to all non-overlapping pairs of particles. The application of an operator to particles $n, n+1$ makes use of the coefficients in $\lambda^{[n-1,n,n+1]}$ and $\Gamma^{[n,n+1]}$, without changing the coefficients in $\lambda^{[n-1,n+1]}$ (see figure 4.2).

Since only the matrices specific to a given pair of particles are changed during the update, we can update adjacent pairs of non-overlapping particles independently. This independence means that we can perform all the updates in parallel, which we will make use of in section 5.4.

4.4.3 Orthogonality of the updated Schmidt vectors

As we saw above, updating the state with a two-particle operator changes the Schmidt vectors for the $[1\ldots n]$ and $[n+1\ldots N]$ subsystems. Moreover, since the Schmidt vectors for the larger subsystems $[1\ldots n+1]$ and $[n\ldots N]$ are given in terms of those for the smaller systems, these Schmidt vectors have also changed. It is important that the new sets of Schmidt vectors stay orthonormal.

To check whether the updated Schmidt vectors are orthonormal, consider first the vectors for $[1\ldots n]$ given in equation (4.29). Taking the inner product and using the fact that the Schmidt vectors for the $[1\ldots n-1]$ subsystem are unchanged and therefore orthonormal, we find

$$\begin{aligned}\Big\langle \overline{\Phi}^{[1\ldots n]}_{\beta'} \Big| \overline{\Phi}^{[1\ldots n]}_{\beta} \Big\rangle &= \sum_{i\alpha i'\alpha'} \Big(\overline{\Gamma}^{[n]i'}_{\alpha'\beta'}\Big)^* \lambda^{[n-1]}_{\alpha'} \lambda^{[n-1]}_{\alpha} \overline{\Gamma}^{[n]i}_{\alpha\beta} \underbrace{\Big\langle \Phi^{[1\ldots n-1]}_{\alpha'} \Big| \Phi^{[1\ldots n-1]}_{\alpha} \Big\rangle}_{\delta_{\alpha'\alpha}} \underbrace{\langle i'|i\rangle}_{\delta_{i'i}} \\ &= \sum_{i\alpha} \big(L_{(i\alpha),\beta'}\big)^* L_{(i\alpha),\beta} = \big(L^\dagger L\big)_{\beta'\beta} = \delta_{\beta'\beta}, \end{aligned} \tag{4.32}$$

due to the unitarity of the singular value decomposition. A similar calculation applies to the vectors for $[n+1\ldots N]$ in equation (4.30).

For subsystems that contain both updated particles we get a different situation. For the

$[1 \ldots n+1]$ subsystem we have

$$\begin{aligned}\left|\overline{\Phi}_{\gamma}^{[1\ldots n+1]}\right\rangle &= \sum_{ij}\sum_{\alpha\beta} \lambda_{\alpha}^{[n-1]}\overline{\Gamma}_{\alpha\beta}^{[n]i}\overline{\lambda}_{\beta}^{[n]}\overline{\Gamma}_{\beta\gamma}^{[n+1]j}\left|\Phi_{\alpha}^{[1\ldots n-1]}\right\rangle\left|ij\right\rangle \\ &= \sum_{ij}\sum_{\alpha\beta} L_{(i\alpha),\beta}\sigma_{\beta}R^{*}_{(j\gamma),\beta}\frac{1}{\lambda_{\gamma}^{[n+1]}}\left|\Phi_{\alpha}^{[1\ldots n-1]}\right\rangle\left|ij\right\rangle. \end{aligned} \tag{4.33}$$

If the sum over β included all Schmidt coefficients, we would get elements from the matrix product $L\Sigma R^{\dagger}$, which by definition equals the expansion matrix Θ (see equation (4.27)). As long as the truncated coefficients are small this will be approximately true:

$$\left|\overline{\Phi}_{\gamma}^{[1\ldots n+1]}\right\rangle \approx \sum_{ij}\sum_{\alpha}\Theta_{\alpha\gamma}^{ij}\frac{1}{\lambda_{\gamma}^{[n+1]}}\left|\Phi_{\alpha}^{[1\ldots n-1]}\right\rangle\left|ij\right\rangle. \tag{4.34}$$

Thus the inner product is

$$\begin{aligned}\left\langle\overline{\Phi}_{\gamma}^{[1\ldots n+1]}|\overline{\Phi}_{\gamma'}^{[1\ldots n+1]}\right\rangle &\approx \sum_{iji'j'}\sum_{\alpha\alpha'}\left(\Theta_{\alpha\gamma}^{ij}\right)^{*}\Theta_{\alpha'\gamma'}^{i'j'}\frac{1}{\lambda_{\gamma}^{[n+1]}\lambda_{\gamma'}^{[n+1]}}\left\langle ij|i'j'\right\rangle\left\langle\Phi_{\alpha}^{[1\ldots n-1]}|\Phi_{\alpha'}^{[1\ldots n-1]}\right\rangle \\ &= \sum_{ij}\sum_{\alpha}\left(\Theta_{\alpha\gamma}^{ij}\right)^{*}\Theta_{\alpha\gamma'}^{ij}\frac{1}{\lambda_{\gamma}^{[n+1]}\lambda_{\gamma'}^{[n+1]}}. \end{aligned} \tag{4.35}$$

A glance at the definition of Θ in (4.25) reveals that this latest contraction yields a large expression involving the matrix elements of the applied two-particle operator as well as the previous Γ-matrices for the two particles. If the operator $\hat{U}$ is unitary we will find that this large expression involves the computation

$$\sum_{ij}\left(U_{k'l'}^{ij}\right)^{*}U_{kl}^{ij} = \left(U^{\dagger}U\right)_{(kl),(k'l')} = \delta_{kk'}\delta_{ll'}. \tag{4.36}$$

The remaining part of the expression can then seen to be equal to the inner product between the previous Schmidt vectors, which are orthonormal:

$$\left\langle\overline{\Phi}_{\gamma'}^{[1\ldots n+1]}|\overline{\Phi}_{\gamma}^{[1\ldots n+1]}\right\rangle \approx \left\langle\Phi_{\gamma'}^{[1\ldots n+1]}|\Phi_{\gamma}^{[1\ldots n+1]}\right\rangle = \delta_{\gamma'\gamma}. \tag{4.37}$$

A similar calculation applies to the $[n \ldots N]$ subsystem.

This shows that if the applied operator is unitary the updated vectors will be approximately orthonormal, up to the truncated Schmidt coefficients. While small nonorthogonalities will inevitably be introduced, this is not a big problem, as an update on the neighboring set of particles will force the vectors to be orthonormal again. A bigger problem, which we consider below, appears when the operators are not unitary.

4.5 Imaginary time-development

One nice thing about being able to do time-development in quantum mechanics is that it almost automatically allows you to find the ground state of the system of interest. This trick consists of

performing time-development along the imaginary time axis instead of the real axis, also known as a Wick rotation of time.

To see how this works, consider again the time-development operator from section 2.4. Writing it out in the energy basis and switching from t to $-it$, we can see that the operator transforms as

$$\sum_n e^{-iE_n t} |n\rangle\langle n| \to \sum_n e^{-E_n t} |n\rangle\langle n| . \tag{4.38}$$

Consider a state $|\psi\rangle$ expanded in the energy basis as $\sum_n \psi_n |n\rangle$. When the above operator acts on such a state, we obtain

$$|\psi(t)\rangle = N(t) \sum_n e^{-E_n t} \psi_n |n\rangle = e^{-E_0 t} N(t) \sum_n e^{-(E_n - E_0)t} \psi_n |n\rangle , \tag{4.39}$$

where we have introduced a normalization factor to account for the fact that the norm of the state changes due to the time-development operator no longer being unitary. Pulling the $e^{-E_0 t}$ into the normalization factor we see that the result of the imaginary time-evolution is to exponentially suppress the excited states, leaving the ground state untouched. The state will therefore converge to the ground state, with the speed of convergence determined by the slowest exponential factor $e^{-(E_1 - E_0)t}$.

Implementing this procedure in TEBD amounts to applying nonunitary two-particle operators $e^{-\hat{H}_n \delta t}$ to the state, in the same way as outlined above. As we saw in section 4.4.3, this leads to nonorthogonalities in the updated Schmidt vectors. As long as the timestep is small, this will actually not be a big problem because the operator is close to unity. However, we want to leave the possibility open to consider longer timesteps.

Consider therefore the procedure for a single timestep in imaginary time. This consists of a series of alternating applications of the operators $e^{-\hat{H}_{\text{odd}} \delta t}$ and $e^{-\hat{H}_{\text{even}} \delta t}$. These in turn consist of a "sweep" along the chain, applying operators to each pair of particles. Assume we are starting with $\hat{H}_{\text{odd}}$ and particles 1 and 2. After updating these two particles we should turn to the next term in $\hat{H}_{\text{odd}}$, involving particles 3 and 4. For this we need the decomposition

$$|\psi\rangle = \sum_{ij} \sum_{\alpha\beta\gamma} \lambda^{[2]}_\alpha \Gamma^{[3]i}_{\alpha\beta} \lambda^{[3]}_\beta \Gamma^{[4]j}_{\beta\gamma} \lambda^{[4]}_\gamma \left|\Phi^{[1,2]}_\alpha\right\rangle |ij\rangle \left|\Phi^{[5\ldots N]}_\gamma\right\rangle , \tag{4.40}$$

but the $\left|\Phi^{[1,2]}\right\rangle$ vectors are no longer orthonormal, which we rely on for applying the operator correctly. However, the neighboring decomposition given by

$$|\psi\rangle = \sum_{ij} \sum_{\alpha\beta\gamma} \lambda^{[1]}_\alpha \Gamma^{[2]i}_{\alpha\beta} \lambda^{[2]}_\beta \Gamma^{[3]j}_{\beta\gamma} \lambda^{[3]}_\gamma \left|\Phi^{[1]}_\alpha\right\rangle |ij\rangle \left|\Phi^{[4\ldots N]}_\gamma\right\rangle \tag{4.41}$$

has orthonormal Schmidt vectors on both sides: The right-hand vectors are untouched, while the left-hand vectors were newly constructed from a singular value decomposition. If we apply an operator to this pair of particles, the $\left|\Phi^{[1,2]}\right\rangle$ vectors will be reconstructed from a new singular value decomposition, ensuring their orthogonality. From there the decomposition in equation (4.40) will be orthogonal, so we can proceed to update this pair of particles.

This means that we can get around the problems with orthogonality by applying identity operators for the even particles in between applying nonunitary operators to the odd particles

and vice versa. When the sweep is completed, the next step is to perform a similar sweep for the even particles. If we started from the left, we would use the decomposition in equation (4.41). Now, however, the orthogonality of the $\left|\Phi^{[4\ldots N]}\right\rangle$ vectors is ruined by the update of particles $[5,6]$. If we start from the right side, however, the first decomposition to consider is (assuming N is even)

$$|\psi\rangle = \sum_{ij}\sum_{\alpha\beta\gamma} \lambda_{\alpha}^{[N-3]}\Gamma_{\alpha\beta}^{[N-2]i}\lambda_{\beta}^{[N-2]}\Gamma_{\beta\gamma}^{[N-1]j}\lambda_{\gamma}^{[N-1]}\left|\Phi_{\alpha}^{[1\ldots N-3]}\right\rangle|ij\rangle\left|\Phi_{\gamma}^{[N]}\right\rangle, \tag{4.42}$$

where both sets of Schmidt vectors are orthonormal. Thus the next sweep can proceed from the right side and move to the left, still taking care to apply identity operators between the particles.

We conclude that two measures are necessary to work around the nonorthogonalities brought about by nonunitary operators. First, we apply identity operators to the particle pairs between the ones we are sweeping over. Second, we perform the sweeps alternately in the left and right directions. The extra identity operators mean that the process of performing a single sweep no longer consists of independent applications. This means that the updates of during a sweep is no longer independent, so that imaginary time evolution can not be parallelized.

4.6 Expectation values

After obtaining a certain time-developed state of interest, we would like to perform measurements on this state, i.e. calculate expectation values. While general observables that act on the complete system are impossible to represent, some special cases can be calculated efficiently. Here we consider two special cases: *Local observables*, i.e. observables acting on single particles or pairs of neighboring particles, and *product observables*, which act on each particle independently.

These two cases cover all the observables we will need. For instance, to measure the energy of a state we use the Hamiltonian, which is assumed to be a sum of local observables. As the expectation value of a sum is the sum of expectation values, it suffices to consider local observables.

4.6.1 Single-particle observables

The simplest case is measuring properties local to a single particle. Given an operator $\hat{O}$ acting on particle n, we use Schmidt decompositions to isolate the particle in question:

$$|\psi\rangle = \sum_{i\alpha\beta} \lambda_{\alpha}^{[n-1]}\Gamma_{\alpha\beta}^{[n]i}\lambda_{\beta}^{[n]}\left|\Phi_{\alpha}^{[1\ldots n-1]}\right\rangle|i\rangle\left|\Phi_{\beta}^{[n+1\ldots N]}\right\rangle. \tag{4.43}$$

By using this expression and the orthonormality of the Schmidt vector we find for the expectation value of $\hat{O}$:

$$\langle\psi|\,\hat{O}\,|\psi\rangle = \sum_{ij}\sum_{\alpha\beta}\langle i|\,\hat{O}\,|j\rangle\,\lambda_{\beta}^{[n]}\left(\Gamma_{\alpha\beta}^{[n-1]i}\right)^{*}\left(\lambda_{\alpha}^{[n-1]}\right)^{2}\Gamma_{\alpha\beta}^{[n-1]j}\lambda_{\beta}^{[n]}. \tag{4.44}$$

Regarding the $\Gamma^{[n]i}$ as matrices and writing $\Lambda^{[n]}$ for diagonal matrices with the Schmidt coefficient on the diagonal, we can view the sum over α as a matrix product and the sum over β as a trace, so that

$$\left\langle \hat{O} \right\rangle = \sum_{ij} O_{ij} \operatorname{Tr}\left[\left(\Lambda^{[n]}\right)^2 \left(\Gamma^{[n-1]i}\right)^\dagger \left(\Lambda^{[n-1]}\right)^2 \Gamma^{[n-1]j} \right]. \tag{4.45}$$

4.6.2 Two-particle observables

For two-particle operators, a similar expression applies, but this case is more cumbersome as it needs the information from both particles. Starting from equation (4.17) for $|\psi\rangle$, sandwiching the operator $\hat{O}$ acting on particles n and $n+1$, and using the orthonormality of the Schmidt vectors, we obtain

$$\begin{aligned}\left\langle \hat{O} \right\rangle &= \sum_{ijkl} \sum_{\alpha\beta\beta'\gamma} \langle ij| \hat{O} |kl\rangle \, \lambda_\gamma^{[n+1]} \left(\Gamma_{\beta\gamma}^{[n+1]j}\right)^* \lambda_\beta^{[n]} \left(\Gamma_{\alpha\beta}^{[n]i}\right)^* \left(\lambda_\alpha^{[n-1]}\right)^2 \Gamma_{\alpha\beta'}^{[n]k} \lambda_{\beta'}^{[n]} \Gamma_{\beta'\gamma}^{[n+1]l} \lambda_\gamma^{[n+1]} \\ &= \sum_{ijkl} O_{kl}^{ij} \operatorname{Tr}\left[\left(\Lambda^{[n+1]}\right)^2 \left(\Gamma^{[n+1]j}\right)^\dagger \Lambda^{[n]} \left(\Gamma^{[n]i}\right)^\dagger \left(\Lambda^{[n-1]}\right)^2 \Gamma^{[n]k} \Lambda^{[n]} \Gamma^{[n+1]l} \right]. \end{aligned} \tag{4.46}$$

4.6.3 Product observables

The final case to consider is observables that take the form of a tensor product of single-particle observables. Consider the operator

$$\hat{O} = \hat{O}^{[n]} \otimes \hat{O}^{[n+1]} \otimes \ldots \otimes \hat{O}^{[n+k]}, \tag{4.47}$$

acting on particles n to $n+k$. By an expansion similar to equation (4.17), the action of the operator on $|\psi\rangle$ is

$$\begin{aligned}\hat{O}|\psi\rangle = &\sum_{\alpha\gamma} \sum_{i_n i_{n+1} \ldots i_{n+k}} \sum_{\beta_n \beta_{n+1} \ldots \beta_{n+k-1}} \lambda_\alpha^{[n-1]} \Gamma_{\alpha\beta_n}^{[n]i_n} \lambda_{\beta_n}^{[n]} \Gamma_{\beta_n\beta_{n+1}}^{[n+1]i_{n+1}} \ldots \Gamma_{\beta_{n+k-1}\gamma}^{[n+k]i_{n+k}} \lambda_\gamma^{[n+k]} \\ &\times \left|\Phi_\alpha^{[1\ldots n-1]}\right\rangle \left(\hat{O}^{[n]} |i_n\rangle\right) \left(\hat{O}^{[n+1]} |i_{n+1}\rangle\right) \ldots \left(\hat{O}^{[n+k]} |i_{n+k}\rangle\right) \left|\Phi_\gamma^{[n+k+1\ldots N]}\right\rangle. \end{aligned} \tag{4.48}$$

Multiplying with $\langle\psi|$ from the left, we find a trace expression similar to to the ones above, but involving all the particles that $\hat{O}$ acts on:

$$\begin{aligned}\left\langle \hat{O} \right\rangle = &\sum_{i_n i_{n+1} \ldots i_{n+k}} \sum_{j_n j_{n+1} \ldots j_{n+k}} O_{i_n j_n}^{[n]} O_{i_{n+1} j_{n+1}}^{[n+1]} \ldots O_{i_{n+k} j_{n+k}}^{[n+k]} \\ &\times \operatorname{Tr}\Big[\Lambda^{[n+k]} \left(\Gamma^{[n+k]j_{n+k}}\right)^\dagger \Lambda^{[n+k-1]} \ldots \left(\Gamma^{[n]j_n}\right)^\dagger \Lambda^{[n-1]} \\ &\qquad \times \Lambda^{[n-1]} \Gamma^{[n]i_n} \ldots \Gamma^{[n+k]i_{n+k}} \Lambda^{[n+k]} \Big]. \end{aligned} \tag{4.49}$$

This sum goes over $2k$ coefficients for a total of d^{2k} terms, which quickly grows to be prohibitively expensive. However, the assumption that $\hat{O}$ is a tensor product operator means that the summation over each pair of i, j coefficients is independent. We can make use of this to simplify the sum so that it can be calculated even for a large number of particles.

To accomplish this we use the linearity of the trace to move the $\hat{O}$ matrix elements and

the sums inside the trace operation. Then we can use the distributive law for matrices to sum over each particle separately. The resulting procedure looks as follows: We start with an initial matrix equaling the central part of the matrix to sum over:

$$E^0 = \left(\Lambda^{[n-1]}\right)^2. \tag{4.50}$$

From this we can build the complete matrix iteratively by summing over consecutive pairs of i, j indices. In the first step we include all the matrices and coefficients associated with particle n.

$$E^1 = \Lambda^{[n]} \left[\sum_{ij} O_{ij}^{[n]} \left(\Gamma^{[n]j}\right)^\dagger E_0 \Gamma^{[n]i}\right] \Lambda^{[n]}. \tag{4.51}$$

We then use this matrix as a starting point to multiply the matrices associated with particle $n+1$. In general, having formed a matrix for the particles up to particle $n+l$, we include particle $n+l+1$ in the expression by calculating the matrix given by

$$E^{l+1} = \Lambda^{[n+l+1]} \left[\sum_{ij} O_{ij}^{[n+l+1]} \left(\Gamma^{[n+l+1]j}\right)^\dagger E^l \Gamma^{[n+l+1]i}\right] \Lambda^{[n+l+1]}. \tag{4.52}$$

After including all the k particles we are left with a matrix E^k, which we can trace to obtain the expectation value:

$$\left\langle \hat{O} \right\rangle = \mathrm{Tr}\left(E^k\right). \tag{4.53}$$

Each step in the above procedure consists of about d^2 matrix multiplications, so that the expectation value can be calculated in about $kd^2\chi_{\max}^3$ operations.

4.7 Constructing superpositions

So far we have avoided explicitly constructing any states other than simple product states, rather finding realistic ground states from product states by imaginary time development. This is because the representation of product states is especially simple, while more general states decompose nontrivially. We can however form simple linear combinations of MPS states by thinking about how the representation for such a state would look.

Consider two MPS states $|\psi\rangle$ and $|\phi\rangle$, with expansions given by

$$|\psi\rangle = \sum_{i_1 \ldots i_n} \Gamma_{i_1}^{[1]} \Lambda^{[1]} \Gamma_{i_2}^{[2]} \ldots \Gamma_{i_N}^{[n]} \tag{4.54}$$

$$|\phi\rangle = \sum_{i_1 \ldots i_n} \tilde{\Gamma}_{i_1}^{[1]} \tilde{\Lambda}^{[1]} \tilde{\Gamma}_{i_2}^{[2]} \ldots \tilde{\Gamma}_{i_N}^{[N]}. \tag{4.55}$$

We need an expansion that yields the combined state $|\psi\rangle + |\phi\rangle$:

$$\begin{aligned}|\psi\rangle + |\phi\rangle &= \sum_{i_1 \ldots i_n} \left(\Gamma^{[1]}_{i_1} \Lambda^{[1]} \Gamma^{[2]}_{i_2} \ldots \Gamma^{[N]}_{i_N} + \tilde{\Gamma}^{[1]}_{i_1} \tilde{\Lambda}^{[1]} \tilde{\Gamma}^{[2]}_{i_2} \ldots \tilde{\Gamma}^{[N]}_{i_N} \right) \\ &= \sum_{i_1 \ldots i_n} \overline{\Gamma}^{[1]}_{i_1} \overline{\Lambda}^{[1]} \overline{\Gamma}^{[2]}_{i_2} \ldots \overline{\Gamma}^{[n]}_{i_N}. \end{aligned} \tag{4.56}$$

Note first that the $\Gamma^{[1]}$ matrices are k row vectors and the $\Gamma^{[N]}$ matrices are column vectors, while the rest of the matrices are more general matrices. In order to form the sum inside the parentheses, we take inspiration from the following analogous expression:

$$(x_1, x_2) \begin{pmatrix} y_1 & 0 \\ 0 & y_2 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = x_1 y_1 z_1 + x_2 y_2 z_2 \tag{4.57}$$

This indicates that it suffices to make sure that the following holds:

$$\overline{\Gamma}^{[1]}_{i_1} = \left(\Gamma^{[1]}_{i_1}, \tilde{\Gamma}^{[1]}_{i_1} \right), \qquad \overline{\Gamma}^{[N]}_{i_N} = \begin{pmatrix} \Gamma^{[N]}_{i_N} \\ \tilde{\Gamma}^{[N]}_{i_N} \end{pmatrix}, \tag{4.58}$$

$$\overline{\Lambda}^{[1]} \overline{\Gamma}^{[2]}_{i_2} \ldots \overline{\Lambda}^{[N]} = \begin{pmatrix} \Lambda^{[1]} \Gamma^{[2]}_{i_2} \ldots \Lambda^{[N]} & 0 \\ 0 & \tilde{\Lambda}^{[1]} \tilde{\Gamma}^{[2]}_{i_2} \ldots \tilde{\Lambda}^{[N]} \end{pmatrix}. \tag{4.59}$$

The two first equations can be regarded as definitions, while the last can be ensured by putting all the middle matrices on block diagonal form:

$$\overline{\Lambda}^{[n]} = \begin{pmatrix} \Lambda^{[n]} & 0 \\ 0 & \tilde{\Lambda}^{[n]} \end{pmatrix}, \qquad \overline{\Gamma}^{[n]}_{i_n} = \begin{pmatrix} \Gamma^{[n]}_{i_n} & 0 \\ 0 & \tilde{\Gamma}^{[n]}_{i_n} \end{pmatrix}. \tag{4.60}$$

This then gives us an expansion of the combined states. It is still not a completely correct expansion, since it is not normalized and the Schmidt vectors are not orthonormal. This can however be corrected by applying identity operators to all pairs of particles, first from left to right and then from right to left. This reconstruct the Schmidt vectors using singular value decompositions, which ensures their orthonormality.

Chapter 5

Implementation

We implemented TEBD in C++, using the *Armadillo* library (Sanderson, 2013) for array storage and linear algebra. This library provides classes such as `cx_matrix` representing complex matrices and `vec` for real vectors. Objects from these classes contain a pointer to a flat array containing the numbers in column-major order (for Fortran compatibility).

Matrix multiplication is supported by operator overloading, so that objects from Armadillo classes can be multiplied together with simple notation. Matrix expressions are encoded using the technique of *expression templates*, which allows a natural notation without sacrificing efficiency. For instance, a common operation is to multiply a matrix `A` with a diagonal matrix with diagonal stored in the vector `b`. This matrix product can be computed as `A*diagmat(b)`. One might expect this expression to inefficiently generate the actual diagonal matrix with lots of zeros and perform a full matrix multiplication between the two matrices. Expression templates, however, ensure that it is compiled into efficient code which takes the diagonal structure of `diagmat(b)` into account.

Linear algebra operations such as diagonalization and singular value decomposition are supported. All such operations compile to appropriate calls to the time-honored `LAPACK` and `BLAS` libraries.

While the theoretical description of the TEBD method used one-based indexing, as is common in mathematics, in C++ one typically uses zero-based indexing. This means for instance that all sums of the form $\sum_{i=1}^{N}$ will change into $\sum_{i=0}^{N-1}$. The translation between the conventions is usually trivial.

When structuring the code we make heavy use of classes and inheritance. A class is essentially a data structure equipped with a collection of *methods*, or functions that work on the data. Instantiating a class yields an *object*, which contains the data specific to that object and has behavior defined by the methods. Classes can inherit from other classes, which means that they contain all the functionality of the parent class and can add or override functionality. This eases code reuse. It is sometimes useful to define a class that specifies that certain functionality is present without implementing it. Such a class is called an *abstract class*, and the compiler enforces that one can not instantiate it, as it is not completely implemented. To complete the implementation one then writes a subclass which inherits from the abstract class, and can be used in place of it.

The implementation of our simulation code consists of three main families of classes, which we present in the following sections.

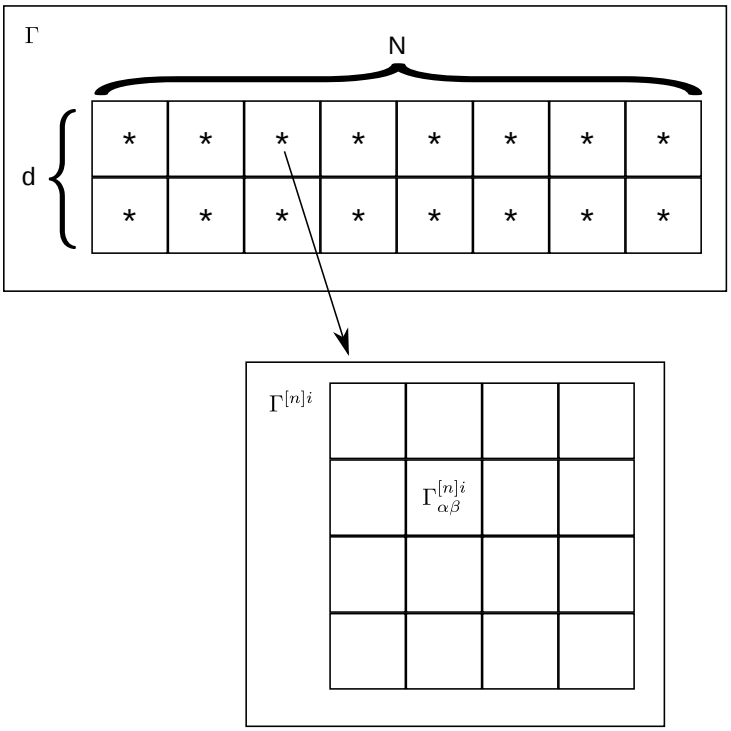


Figure 5.1: Illustration of the `gamma` array. Asterisks indicate pointers. One of the pointers is illustrated by an arrow pointing to a typical matrix $\Gamma^{[n]i}$.

5.1 MPS

The decomposition of the current state $|\psi(t)\rangle$ is represented using the `MPS` class. It is responsible for keeping track of the state as well as applying the two-particle operators that generate time-evolution (see section 4.3).

5.1.1 Representation

The MPS class uses two arrays to represent the decomposition, each again containing other arrays:

- `gamma` is a two-dimensional $N \times d$ array containing the complex $\Gamma^{[n]i}$ matrices (see figure 5.1),
- `lambda` is a one-dimensional $N + 1$-long array containing the real $\lambda^{[n]}$ vectors.

Note that the length of the `lambda` array is $N + 1$ instead of $N - 1$. This is because we add an additional vector of length 1 and value 1 at each end of the array. This allows us to bring certain special cases at the ends of the spin chain into the general form. For instance, the expectation value of an operator $\hat{O}$ acting on the first particle is given by

$$\left\langle \hat{O} \right\rangle = \sum_{ij\beta} O_{ij} \lambda^{[1]}_{\beta} \left(\Gamma^{[1]i}_{\beta} \right)^{*} \Gamma^{[1]i}_{\beta} \lambda^{[1]}_{\beta}. \tag{5.1}$$

To bring this on the general form (4.45) we add a $\lambda^{[0]}$ with one entry equal to 1, and give the $\Gamma^{[0]i}$ an additional one-dimensional index (having no effect). This gives us

$$\left\langle \hat{O} \right\rangle = \sum_{ij\alpha\beta} O_{ij} \lambda^{[1]}_{\beta} \left(\Gamma^{[1]i}_{\alpha\beta} \right)^{*} \left(\lambda^{[0]}_{\alpha} \right)^{2} \Gamma^{[1]i}_{\alpha\beta} \lambda^{[1]}_{\beta} = \sum_{ij} O_{ij} \operatorname{Tr} \left[\left(\Lambda^{[1]} \right)^{2} \left(\Gamma^{[1]i} \right)^{\dagger} \left(\Lambda^{[0]} \right)^{2} \Gamma^{[1]i} \right], \quad (5.2)$$

which is similar to the general case. Thus the loops over the N particles do not have to distinguish special cases.

Such expectation-value calculations are provided as methods of the class. For instance, representing a two-particle operator acting on particles n and $n+1$ as a $d^2 \times d^2$ complex matrix, its expectation value can be computed by the following method, which can be compared with equation (4.46):

```
double MPS::expectation2(const cx_mat &op, const ix n) const
{
  cx_double ret = 0;
  for (ix i = 0; i < d; i++) {
    for (ix j = 0; j < d; j++) {
      for (ix k = 0; k < d; k++) {
        for (ix l = 0; l < d; l++) {
          ret += trace(diagmat(lambda(n))*diagmat(lambda(n))*
              gamma(n, i)*diagmat(lambda(n+1))*gamma(n+1, j)*
              diagmat(lambda(n+2))*diagmat(lambda(n+2))*
              gamma(n+1, k).t()*diagmat(lambda(n+1))*
              gamma(n, l).t())*op(d*l+k, d*i+j);
        }
      }
    }
  }
  return ret.real();
}
```

5.1.2 Initialization

An `MPS` object called `mps` is declared simply as `MPS mps`. This object contains an empty state and must be initialized to describe anything useful. To this end one would typically use one of the two `product` methods, whose signatures are shown below:

```
void MPS::product(const cx_mat &coeff);
void MPS::product(const ix N, cx_vec &coeff);
```

These methods initialize the state as a product of single-particle states, which as we found in section 4.1.1 is simple. The first method takes an $N \times d$ complex matrix where each row gives the single-particle state for each particle. The second method takes a d-dimensional complex vector describing a single-particle state $|s\rangle$ and initializes the state as $|\psi\rangle = |s\rangle^{\otimes N}$. An additional parameter `N` is required to determine the number of particles.

5.1.3 Applying operators

As described in section 4.4, the crucial part of time-development is applying two-particle operators to the state. This process is implemented by a method with the following signature:

```
double MPS::applyDoubleOp(const cx_mat &U, const ix l);
```

This method takes a $d^2 \times d^2$ matrix representing the two-particle operator to apply and the position of the first of the two particles. Using this, it calculates the Θ matrix of the updated state and stores it in a matrix `theta`, contained in the `MPS` object. This calculation involves heavily nested loops. As the matrix involves the index-combinations introduced in section 2.3.1, these combinations are calculated along the way. The core part of the method is shown below:

```
for (ix k = 0; k < d; k++) {
  for (ix l = 0; l < d; l++) {
    cx_mat mprod = diagmat(lambda(n)) * gamma(n, k)
                 * diagmat(lambda(n+1)) * gamma(n+1, l)
                 * diagmat(lambda(n+2));
    for (ix i = 0; i < d; i++) {
      for (ix j = 0; j < d; j++) {
        for (ix alpha = 0; alpha < alsize; alpha++) {
          for (ix gam = 0; gam < gamsize; gam++) {
            theta(alsize*i+alpha, gamsize*j+gam) +=
                  U(d*i+j, d*k+l)*mprod(alpha, gam);
          }
        }
      }
    }
  }
}
return resplit(n);
```

Note that the loop over k and l is prioritized over the ij loop. This is because the product

$$\sum_{\beta} \lambda_{\alpha}^{[n-1]} \Gamma_{\alpha\beta}^{[n]k} \lambda_{\beta}^{[n]} \Gamma_{\beta\gamma}^{[n+1]l} \lambda_{\gamma}^{[n+1]} \tag{5.3}$$

can be seen as the elements of a matrix product

$$\Lambda^{[n-1]} \Gamma^{[n]k} \Lambda^{[n]} \Gamma^{[n+1]l} \Lambda^{[n+1]}, \tag{5.4}$$

which can be calculated for each k, l. Starting by computing this matrix for each k and l allows us to reuse its coefficients in the subsequent iterations, saving time.

Having calculated Θ, the method then calls the `resplit` method, which performs a singular value decomposition of the Θ matrix to obtain the updated Γ matrices and Schmidt coefficients. It then throws away the lowest Schmidt coefficients when the number of coefficients is too high, a task which is greatly helped by the fact that the vector of Schmidt coefficients returned from the Armadillo function `svd` is sorted from high to low. The truncated Schmidt coefficients are squared, summed and returned as an error estimate.

In addition to throwing away Schmidt coefficients when their number is too big, we also remove any Schmidt coefficients that are smaller than 10^{-12} to avoid stability issues from the division by small Schmidt coefficients in equation (4.28).

5.1.4 Saving and loading

The `MPS` class provides saving to and loading from disk. This allows us to find the ground state in one run, and then use the same state as a basis for many different future runs. It also allows snapshots of the state at different running times to be saved, after which expensive offline analyses can be performed.

Saving the MPS to disk is done using the `save_file` method, which makes use of the built-in armadillo support for disk I/O to perform its task in a very simple way:

```
void MPS::save_file(const char* name) const
{
        std::ofstream ofile(name);
        gamma.save(ofile);
        lambda.save(ofile);
        ofile.close();
}
```

The `load_file` method simply mirrors the above and performs some initialization of other variables based on the loaded arrays.

5.2 Hamiltonian

The `Hamiltonian` class represents the Hamiltonian of the system and is responsible for keeping track of the system's time-dependence.

5.2.1 Representation

The representation of the terms in the Hamiltonian is somewhat peculiar in order to optimize it for the problem at hand. The main data contained in the object is a $d^2 \times d^2$ complex matrix `twopart` describing all of the nearest-neighbor terms H_n, which are all assumed to be equal.

In order to accommodate exceptions to translational symmetry (such as impurities and boundary terms), there is an N-long array of $d^2 \times d^2$ matrices `exceptions` containing any exceptions to this, as well as an N-long array of booleans `isException` describing whether a particular pair of particle uses this exception array. The `add_exception` method can be used to add an exception at a particular pair of particles.

We also want to support time-dependent terms in the Hamiltonian. To this end, the Hamiltonian object stores a linked list `timedep` of objects from an abstract class `Timedep`, which acts as an interface for time-dependent terms. The `Timedep` class only supports time-dependencies consisting of a sum of single-particle terms, although support for more complex terms is possible. An abstract subclass of `Timedep` called `ConstField` describe time-dependent terms that act equally on all particles. This class is then again subclassed to describe concrete time-dependencies. Examples of this include `Laser`, which describes a sinusoidal oscillation term, and `Linear`, which describes a linearly increasing term.

5.2.2 Initialization

When creating a Hamiltonian object one calls its constructor, which creates a zero Hamiltonian for the given system size:

```
Hamiltonian::Hamiltonian(ix N, ix d);
```

In order to modify the `twopart` matrix and the exception arrays one uses predefined methods such as `add_ising` and `add_heisenberg`. These methods add terms appropriate for the indicated spin models. The method `add_magnet` can be used to add a uniform magnetic field in a given direction.

As an example, the `add_ising` method is shown below. It takes the direction of the interaction (X, Y or Z) and a coupling constant, calculates the spin matrix along the given direction and adds the appropriate term to the `twopart` matrix. The tensor product of two matrices is provided by the `kron` function from Armadillo.

```
void Hamiltonian::add_ising(enum Dir dir, double J)
{
        cx_mat s = spin_matrix(dir, d);
        twopart += -J*kron(s, s);
}
```

Time-dependent terms can be added similarly. For example, to add the laser term given in equation (3.25) one calls the following method:

```
void Hamiltonian::add_laser(enum Dir dir, double A, double om);
```

This takes care of constructing the appropriate `Timedep` object and adding it to the list of time-dependent terms.

Taken together these methods allow us to assemble the Hamiltonian from different parts. For instance, to create an Ising model in a transverse magnetic field we make the following calls:

```
Hamiltonian ham(N, d);
ham.add_ising(Z, Jz);
ham.add_magnet(X, h);
```

This can e.g. be used to perform imaginary time-development and find the ground state. When this is complete, we can add more terms to experiment with the system, such as adding a laser using the above method.

5.3 Sweeper

As we saw in section 2.7, there are several different ways to decompose the time-development operator into manageable pieces. To accommodate different decompositions we write a class for each of them. These classes are subclasses of the abstract class `Sweeper`, which defines the interface for a decomposition. Any sweeper has a reference to a particular Hamiltonian, which supplies the two-particle terms that should be exponentiated to obtain parts of the time-development operator, as in section 4.3.

The simplest implementation of a `Sweeper` is the second-order Trotter decomposition, implemented in the class `Trotter2`. An object from this class is initialized with the following constructor:

```
Trotter2(Hamiltonian& ham, const double dt, const bool imag=false);
```

The constructor takes a reference to the Hamiltonian of the system, a timestep and optionally a boolean indicating whether the object should perform real or imaginary time-development. It then goes on to calculate the exponentiated operators for the even and odd terms using the following method:

```
void Trotter2::calc_exps()
{
        cx_double mul = -dt;
        if (!imag) mul *= cx_double(0, 1);
        for (ix n = 0; n < ham.N-1; n += 2) {
                // Even terms
                evens(n/2) = get_exp(n, 0.5*mul);
        }
        for (ix n = 1; n < ham.N-1; n += 2) {
                // Odd terms
                odds(n/2) = get_exp(n, mul);
        }
}
```

The exponentiated operators are stored in two separate arrays of matrices. The actual exponentiation is done through a helper method in the `Sweeper` class. A call to `get_exp(n, mul)` returns the matrix $\exp[\text{mul} \cdot H_n(t)]$. This is done using eigendecomposition of the Hermitean matrix H_n, using the Armadillo function `eig_sym`:

```
cx_mat Sweeper::get_exp(ix n, cx_double mul) const
{
  cx_mat U; vec D;
  eig_sym(D, U, ham.get_term(n));
  return U*diagmat(exp(mul*D))*U.t();
}
```

A single time step can then be performed on a state by calling the following method:

```
double Trotter2::timeStep(MPS *mps)
{
        double err = 0;
        if (ham.timeStep(dt))
                calc_exps();
        err += sweep(evens, even, mps);
        err += sweep(odds,  odd,  mps);
        err += sweep(evens, even, mps);
        return err;
}
```

First, the Hamiltonian is notified of the timestep. This returns a boolean indicating whether the Hamiltonian changed (i.e. the Hamiltonian is time-dependent), which means that the operator exponents must be recalculated. After this the actual sweeping is performed with the help of a method `sweep` from the abstract base class `Sweeper`.

In this particular Trotter decomposition, one first performs a sweep of the even particles, using the `evens` array containing matrices of the form $\exp(-\frac{1}{2}iH_{2n}\text{dt})$, so that the even particles are developed for half a time step. Then the odd particles are updated using the `odds` array,

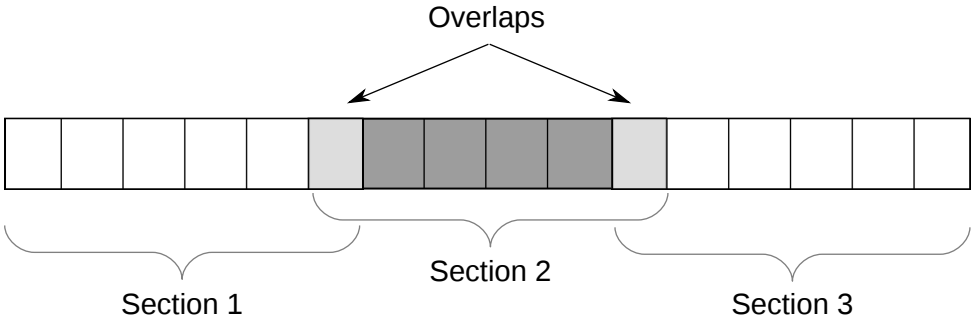


Figure 5.2: Overlap between subchains

containing matrices of the form $\exp(-iH_{2n+1}\mathrm{dt})$, so that these particles are developed for a full time step. Finally, the even sweep is repeated.

The `sweep` method from the `Sweeper` class takes care of alternating the directions of the sweep, as described in section 4.5. After deciding which direction to sweep, it again calls the `sweep` method of the `MPS` class, which in turn calls `applyDoubleOp` on each of the operators in the given array.

5.4 Parallelization

As noted in section 4.4, the even-odd decomposition has the effect that the two-particle operators during each sweep can be applied independently. This enables us to speed up the simulation by applying several two-particle operators in parallel during each time step.

The `ParMPS` subclass of the `MPS` class implements this parallelization, using the *Message Passing Interface* (MPI) for multi-processing and communication between processes. The subclass approach allows the client program to choose between a parallel implementation and a sequential one, as well as enabling a lot of code sharing between the two implementations.

Not all features of the `MPS` class can be implemented to run in parallel by the `ParMPS` class. The most obvious of these is imaginary time-development, as updating the particle pairs is not independent in this case (see section 4.5). However, since the most time-consuming simulations involve real time-development, we can calculate the ground state on a single processor using the `MPS` class. The ground state can the be saved to disk and subsequently loaded into the `ParMPS` class to continue with the more demanding simulation.

5.4.1 Representation

Given N particles to be simulated on M processors, with $M < N/2$, we divide the chain into M overlapping sections of approximately equal size, as shown in figure 5.2. With this overlap of one particle, every pair of particles is completely contained in exactly one section, and can therefore be updated by the process associated to this section.

The `calc_locs` method determines exactly how the chain is split into subchains. The index of the first and last particles belonging to the current process needs to be calculated in a manner which is consistent across all the process. Doing this correctly requires some tedious logic.

The `gamma` and `lambda` arrays are then initialized to store the Γ and λ coefficients specific to the current subchain. These arrays need to be indexed correctly when sweeping over even and odd particles. The `sweep` method is therefore overridden to ensure this.

5.4.2 Communication

After a sweep is performed, the matrices describing particles in the overlaps have been updated by one of the processes responsible for the particle, but not by the other. The updated matrices must therefore be communicated to the other process so that a consistent description is maintained.

Sending and recieving multiple messages on both sides of the subchain requires some care. One can imagine deadlock scenarios, where processes wait for messages that will never be sent. To avoid this possibility we use asynchronous calls to send off all the necessary messages first, and then wait for them to arrive at a later point.

The `update_matrices` method performs the necessary communication after each sweep. To do this correctly it needs to know whether to send the matrices to the neighboring process, or to recieve from it. This information is contained in the `sendLeft` and `sendRight` booleans, which are passed as parameters to the method. Also, if the subchain is at the far left of the chain, indicated by the `isLeft` boolean, no communication needs to happen to the left. Similarly, if the `isRight` boolean is set, no communication happens to the right.

Since the matrix sizes can change during a sweep, the new matrix sizes must be communicated before any data can be sent. The code that sends or recieves updated sizes from the process to the left of the current process is shown below.

```
if (!isLeft) {
  if (sendLeft) {
        MPI_Isend(&sizeLeft, 1, MPI_UNSIGNED_LONG, comm_rank - 1,
                  mktag(left, comm_rank, 0), comm, &reqs[n_reqs++]);
  } else {
        MPI_Irecv(&sizeLeft, 1, MPI_UNSIGNED_LONG, comm_rank - 1,
                  mktag(right, comm_rank - 1, 0), comm, &reqs[n_reqs++]);
  }
}
// Similar for right hand side
wait();
```

After determining whether to send or recieve data, an MPI call is issued to do the actual communication. To ensure that the right message goes to the right process, the `mktag` helper method creates a unique message tag based among others on the direction of the message (left to right or right to left) and the sending process (`comm_rank`).

When all the messages are sent, the `wait` helper method waits until all the messages registered in the `reqs` array have arrived. The matrices can therefore be resized to the new matrix sizes, and the updated data from each matrix can be sent, using code similar to the above code. The `wait` method is then called once again to ensure that all the communication completes before the matrices are used again.

Chapter 6

Results

We saw in chapter 3 that the linear-chain ferromagnet $CoCl_2 \cdot 2\,H_2O$ should be capable of exhibiting magnetic Bloch oscillations. Attempts to verify this by observing the Wannier-Zeeman ladder have so far been unsuccessful (Christensen et al., 2000; Montfrooij et al., 2001). This is partly explained by the fact that the spectral signature of the Wannier-Zeeman ladder is weak (Shinkevich and Syljuåsen, 2012).

As a more direct way of observing magnetic Bloch oscillations, Shinkevich and Syljuåsen (2013) proposed using lasers to excite the actual oscillations. Numerical simulations had the desired results, but the simulations were limited to small systems or a small number of domain walls, which reduces the confidence that the simulations are realistic.

Here we use our implementation of the TEBD method to study laser-excited Bloch oscillations in a large system with no restriction on the number of domain walls. The material is modeled as a linear N-site Heisenberg chain with a magnetic field along the z-direction, described by the Hamiltonian in equation (3.12), which is restated here for reference:

$$\hat{H}(t) = \hat{H}_0 + \hat{H}_m + \hat{H}_l(t), \tag{6.1}$$

$$\begin{aligned}\hat{H}_0 = &-J_z \sum_{j=1}^{N-1} \hat{S}_j^z \hat{S}_{j-1}^z - J_a \sum_{j=1}^{N-1} \left(\hat{S}_j^+ \hat{S}_{j+1}^+ + \hat{S}_j^- \hat{S}_{j+1}^- \right) \\ &-J_\perp \sum_{j=1}^{N-1} \left(\hat{S}_j^+ \hat{S}_{j+1}^- + \hat{S}_j^- \hat{S}_{j+1}^+ \right),\end{aligned} \tag{6.2}$$

$$\hat{H}_m(h_z) = -h_z \sum_{j=1}^{N} \hat{S}_j^z. \tag{6.3}$$

We use open boundary conditions as appropriate for the TEBD method. The time-dependent term $\hat{H}_l(t)$ in the Hamiltonian describes the lasers we use to excite the system. As in section 3.3, we model a single laser as an oscillating magnetic field, given by

$$\hat{H}_l(t; V, \omega) = -V \cos \omega t \sum_{j=0}^{N} \hat{S}_j^z. \tag{6.4}$$

For two lasers with different frequencies and amplitudes we simply add two terms of the above form.

The coupling constants used are those given in table 3.1, rescaled to units of J_z, and the magnetic field strength will be fixed at $h_z = 0.05 J_z$. We mostly use a system size of $N = 100$ sites, which gives room for several domains even when they get as large as 10 – 12 sites, while still keeping the simulations manageable.

6.1 Verification

In order to verify that our implementation is correct, we performed redundant calculations using different methods to obtain the same answer. By considering small enough systems the exact solution can be calculated by exact diagonalization. For larger systems we use the adiabatic theorem to estimate the ground state in two different ways.

6.1.1 Exact diagonalization

If we restrict the system to consisting of only 10 particles, the dimension of the Hilbert space is $2^{10} = 1024$, which is entirely manageable on a computer. Diagonalizing the 1024×1024 matrix representing the Hamiltonian allows us to find the ground state as well as the time evolution operator. The results of such calculations can be compared with the results of a TEBD calculation for the same system.

We considered the Hamiltonian in equation (6.3) with $N = 10$ particles. In order to compare the TEBD method to exact diagonalization we calculated three different error estimates. The first error estimate is the TEBD truncation error $\epsilon_t(t)$. This is obtained by squaring and summing the Schmidt coefficients that were truncated from the state each time an operator is applied (see section 4.4), to get an estimate of the error from the truncation at each time step. Accumulating these error estimates from all the time steps then gives an indication of the total error from the TEBD approximation.

The next estimate measures how far the TEBD-simulated state is from the exact state at any point in time. This *fidelity error* is defined by

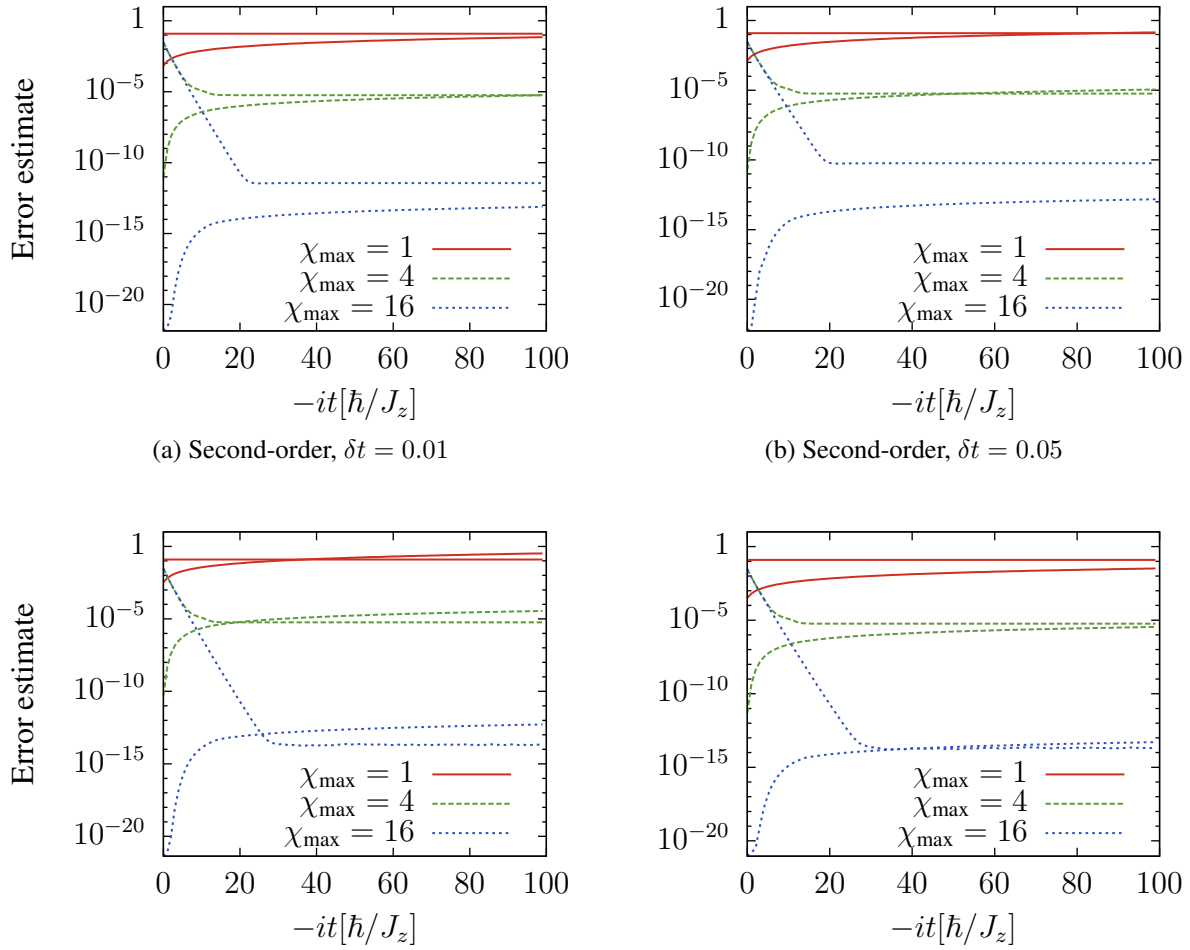
$$\epsilon_f(t) = 1 - |\, \langle \psi(t)_{\text{TEBD}} | \psi(t)_{\text{exact}} \rangle \,|^2. \tag{6.5}$$

Finally, in order to relate the error estimates to something more tangible, we measure the relative error of the magnetization, by

$$\epsilon_m(t) = \frac{|N_\uparrow^{\text{exact}} - N_\uparrow^{\text{TEBD}}|}{N_\uparrow^{\text{exact}}}. \tag{6.6}$$

Imaginary time evolution

We found the ground state of the system by imaginary time evolution from the ferromagnetic state. This was compared with the exact ground state found by diagonalizing the Hamiltonian. The result is shown in figures 6.1a and 6.1b for different values of χ_{max}. The fidelity error decreases with imaginary time, which means that the TEBD state approaches the exact ground state. Meanwhile the truncation error increases from zero as more Schmidt coefficients are truncated from the state.



(a) Second-order, $\delta t = 0.01$

(b) Second-order, $\delta t = 0.05$

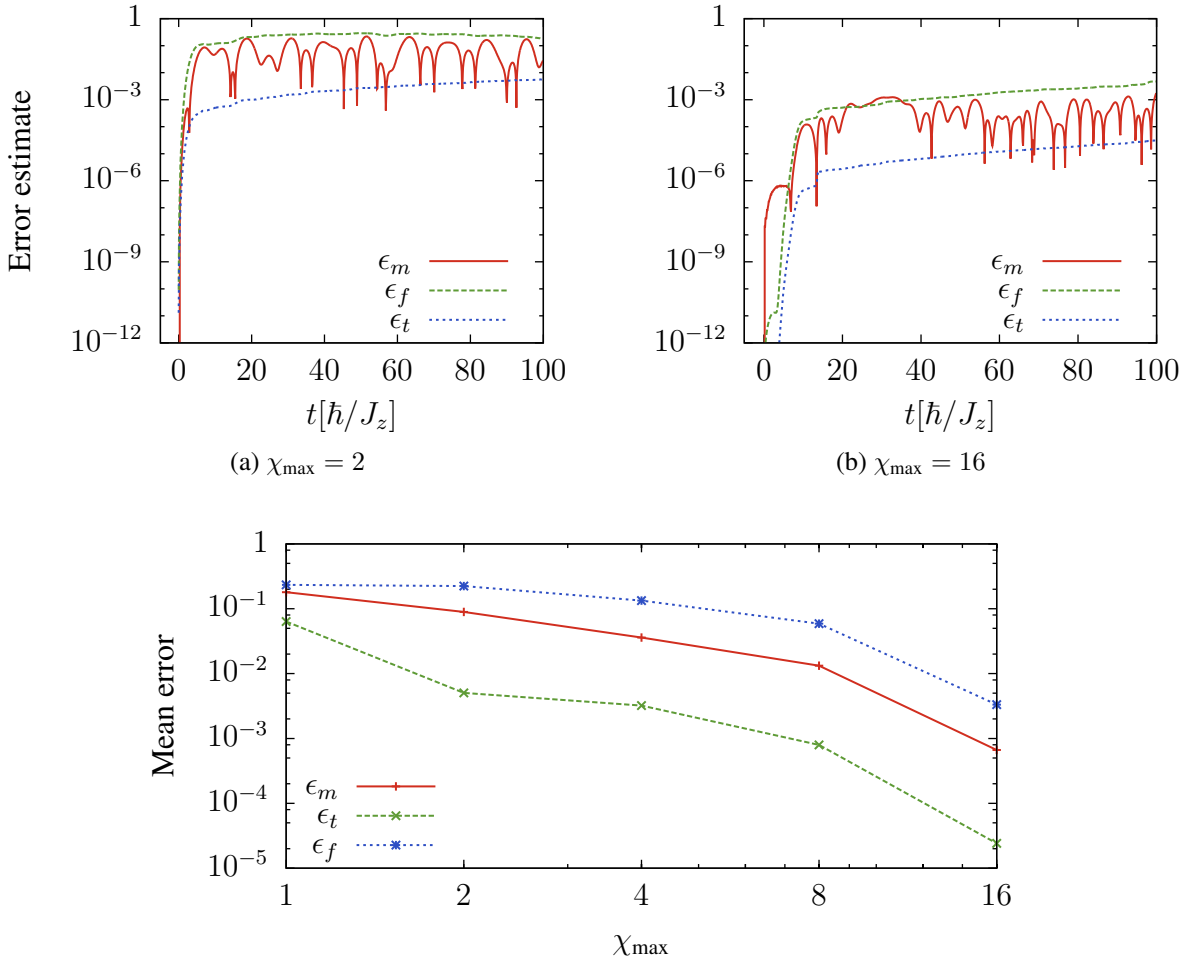
(c) Irrational third-order, $\delta t = 0.01$

(d) Integral third-order, $\delta t = 0.01$

Figure 6.1: Error estimates of the ground state using imaginary time evolution, with varying timesteps and Suzuki-Trotter decompositions. The lines starting at 1 show the fidelity error compared to the exact ground state, while the lines starting from below show the truncation error from the truncated Schmidt coefficients.

The two estimates tend to agree after long times, meaning that the truncation error is a good error estimate. For $\chi_{\max} = 16$ the truncation error underistimates the total error, forming a gap between the two curves. This is probably because the truncation error is so small that the error from the Suzuki-Trotter decomposition dominates, while the truncation error does not take this error source into account. The Suzuki-Trotter error is controlled by the time step, and indeed the fidelity error increases with larger timesteps at $\chi_{\max} = 16$. The truncation error does not react to the larger timestep, so that the gap between the estimates is widened.

We also performed the same tests using third-order Suzuki-Trotter decompositions (see section 2.7). As seen in figures 6.1c and 6.1d, the gap between the two error estimates closes even for large $\chi_{\max}$, which we interpret to mean that the Suzuki-Trotter error again becomes negligible with third-order decompositions.



(a) $\chi_{\max} = 2$

(b) $\chi_{\max} = 16$

(c) Mean values of the error estimates for the last 20 time units, for varying $\chi_{\max}$.

Figure 6.2: Three error estimates for the real time evolution of a Heisenberg model with a magnetic field starting in the ferromagnetic state: Fidelity error, accumulated error from the truncated Schmidt coefficients, and relative error in the magnetization.

Real time evolution

We also performed real time evolution using TEBD and exact diagonalization. For anything to happen, it is necessary to start with an initial state which is not an eigenstate of the Hamiltonian. The ferromagnetic state is one such state which is easy to construct.

In figures 6.2a and 6.2b we show the error estimates from two time-evolutions starting from the ferromagnetic state. Both the truncation error and the fidelity error are good estimates for the relative error of the magnetization, although the truncation error underestimates the error. In order to see the dependence of the error on $\chi_{\max}$, we took the mean value of the error estimates over the last 20 time units, so as to smooth out the fluctuations of the magnetization error. As we can see from figure 6.2c, the errors decrease nicely with increasing $\chi_{\max}$.

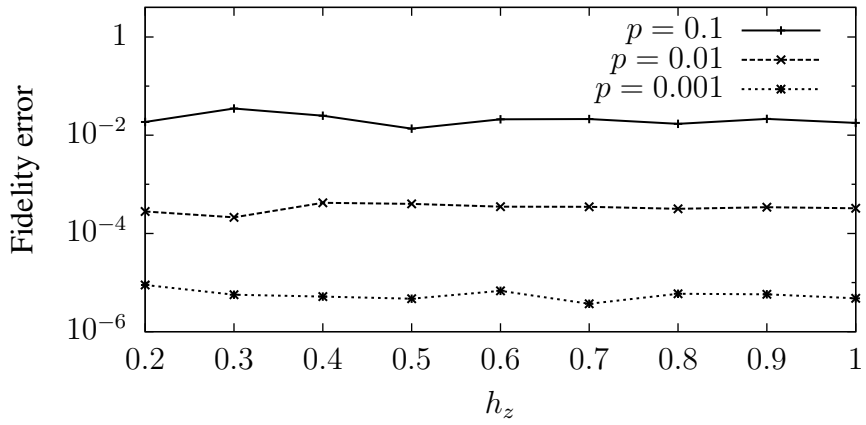


Figure 6.3: The fidelity error between ground states for the Heisenberg model with magnetic field h_z, calculated in two different ways: By imaginary time evolution, and by adiabatic evolution from the $h_z = 0.1$ state. Different lines corresponds to different rates p of increasing h_z.

6.1.2 Adiabatic evolution

In order to validate the implementation for larger systems, we make use of the adiabatic theorem. Consider a Hamiltonian $\hat{H}(\lambda)$ dependent on some parameter λ, with a nondegenerate ground state and a finite energy gap to the first excited state. If we start from the ground state of the Hamiltonian at some value λ_0 and develop the system in time while slowly changing λ, the adiabatic theorem says that the state will adapt to the changing Hamiltonian, remaining approximately in the ground state at all times. The error in this approximation is controlled by the rate that λ changes and the energy gap.

To exploit this fact, we start with finding the ground states for our Heisenberg model with different magnetic fields $0.1J_z,\ 0.2J_z \ldots J_z$, by using imaginary time development. Having found these ground states, we set $h_z = 0.1J_z$ and load the ground state for this Hamiltonian, and start evolving in time while slowly increasing the magnetic field by $h_z = (0.1 + ptJ_z/\hbar)J_z$, so that p determines how quickly the field is increased. Whenever h_z hits one of the values for which we found the ground state, we calculate the fidelity error between the two states. According to the adiabatic theorem, this error will be controlled by the rate p of the magnetic field increase.

Our expectations are confirmed by the result shown in figure 6.3, where the two computations of the ground states agree to a precision controlled by the p parameter.

6.2 Ground state

Having confirmed that our implementation of the TEBD method works well, we turn to the main results. The ground state of the system was found using imaginary time evolution starting from the ferromagnetic state $|\uparrow\rangle^{\otimes N}$. The magnetization and energy stabilized quickly to constant values, as seen in figure 6.4. After the values kept stable for some time the ground state was saved to disk, to be used as a starting point for further calculations.

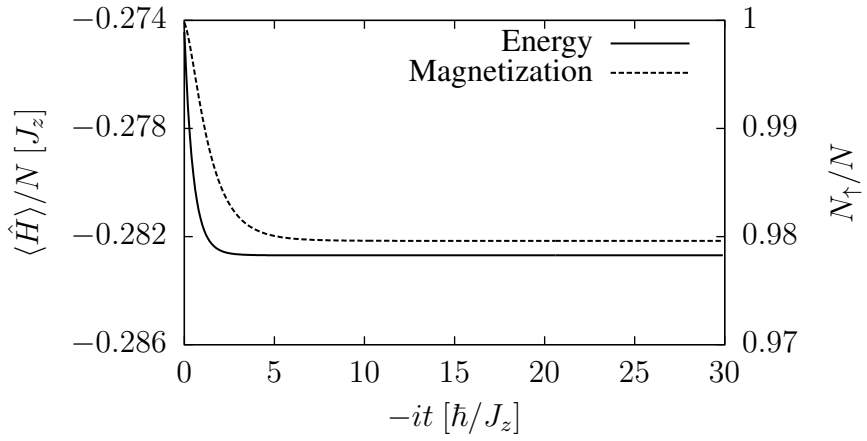


Figure 6.4: Magnetization and energy during imaginary time development to find the ground state of the 100-site system.

6.2.1 Domain distributions

Because of the $\hat{H}_a$ and $\hat{H}_\perp$ terms in the Hamiltonian, the ground state is a superposition containing domains of all sizes with small amplitudes. To investigate the population of domains in the ground state we form the operator

$$\hat{D}_l^j = \left(\hat{S}_{j-1}^z + \frac{1}{2}\right)\left(\hat{S}_j^z - \frac{1}{2}\right)\left(\hat{S}_{j+1}^z - \frac{1}{2}\right)\cdots\left(\hat{S}_{j+l}^z - \frac{1}{2}\right)\left(\hat{S}_{j+l+1}^z + \frac{1}{2}\right). \tag{6.7}$$

When this operator acts on a product state, it has a value of one exactly when the state contains a domain of length l starting at position j, and a value of zero otherwise. Therefore, taking the expectation value of the operator $\hat{D}_l = \sum_{n=2}^{N-l-1} \hat{D}_l^j$ on a general state gives the average number of domains of length l in the state.

We have plotted these expectation values for the ground state in figure 6.5. The domain populations split into two separate exponential functions, for even and odd domain lengths. This can be explained by considering the $\hat{H}_a$ term as a perturbation from the $\hat{H}_z$ term and consider the ground state by perturbation theory from the ferromagnetic state. One will then find that domains of odd length only enters the ground state in pairs, and are therefore effects of higher order than domains of even lengths.

In the single-domain subspace, the only states containing domains of size l are the $|j, l\rangle$ states (see section 3.2.4). Because of this we can regard the $\hat{D}_l^j$ operators as projection operators $|j, l\rangle\langle j, l|$ in the single-domain subspace. The expectation then becomes

$$\left\langle \hat{D}_l \right\rangle_{n,p} = \langle n, p | j, l \rangle \, \langle j, l | n, p \rangle = |\psi_{n,l}(p)|^2. \tag{6.8}$$

The expectation values should therefore be compared with the single-domain wavefunctions. Such a comparison is also shown in figure 6.5. The single-domain wavefunction contains fewer large domains than our ground state. We do not have a good explanation for this.

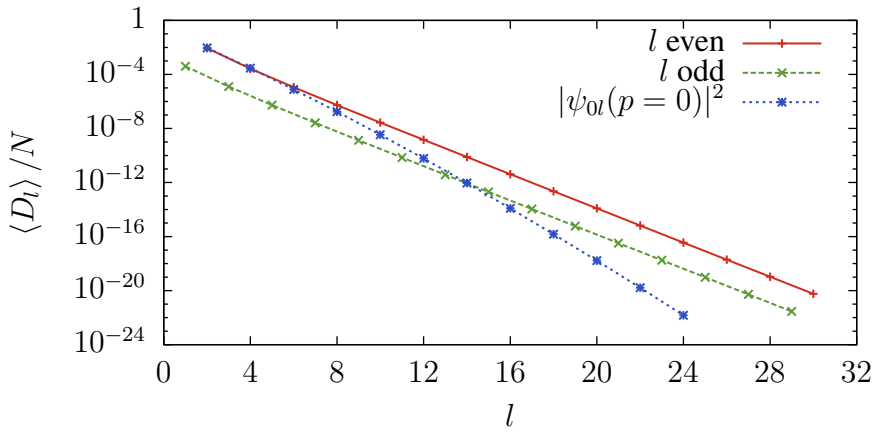


Figure 6.5: Population of domains in the ground state. The populations follow two separate lines (exponential functions), one for even domain lengths and one for odd domain lengths. Even domain lengths have larger populations than odd domain lengths. The single-domain wavefunction agrees with the distribution when l is small and even, but is smaller for large l.

6.3 Study of resonances

Our goal is to use lasers in resonance with the energy levels to excite states in the Wannier-Zeeman ladder. While energy levels for the single-domain approximation can be found analytically (see section 3.2.4), these are not accurate enough to get a proper resonance with realistic laser amplitudes. We therefore performed a search around the approximate energies in order to find the true resonance.

To confirm that the resonance energies are near the single-domain energy levels, we started off by trying many possible frequencies between the $l = 2$ and $l = 14$ energies. Because this was a rough search, the laser amplitude was set at a high value of $V_0 = 0.1J_z$. This ensures that something happens even if we do not use the precise resonance value of the laser frequency.

The results from two such runs are shown in figure 6.6, where we see a markedly bigger effect on the magnetization at resonance. Taking the minimal value of the magnetization is then expected to indicate how close we are to resonance. The minimal value for many different laser frequencies are shown in figure 6.7. It is clear that resonances are achieved at even energy levels, while nothing happens at odd energy levels. As we saw in section 6.2.1, odd-length domains enter the ground state only in pairs. This means that the odd n states with only single domains do not couple to the ground state, explaining the lack of resonance at the odd n energy levels.

It is also clear that the strength of the resonance decreases with higher energy levels, because the coupling to the ground state decreases with energy. Above the 12th excitation it seems like the couplings become too weak to show any resonance with this laser strength and resolution in ω.

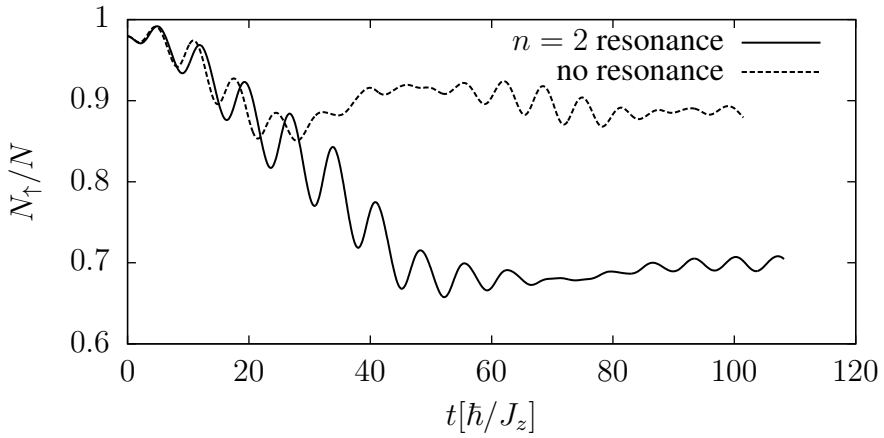


Figure 6.6: Two example runs applying a laser to the material with different frequencies. One laser is at resonance with energy gap to the $n = 2$ state, and one is away from resonance.

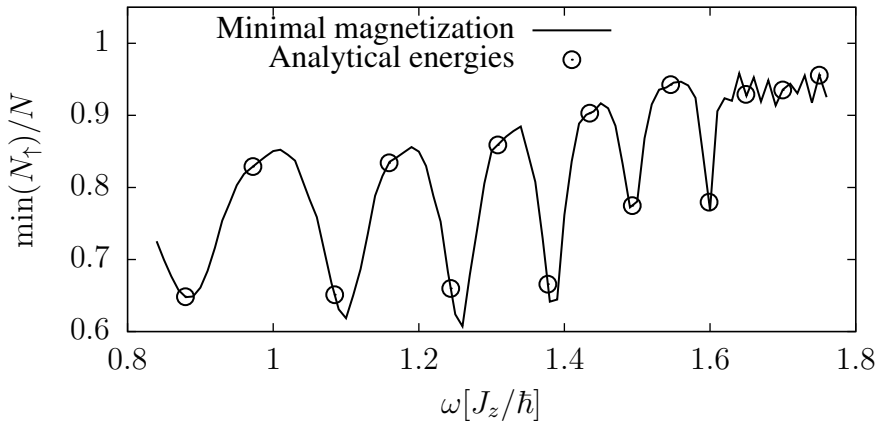


Figure 6.7: Minimal magnetizations for different laser frequencies. The horizontal positions of the circles indicate the approximate theoretical energy levels starting at the $n = 2$ energy. We find resonant behavior near the even energy levels up to $n = 12$. No resonances are found above $n = 12$.

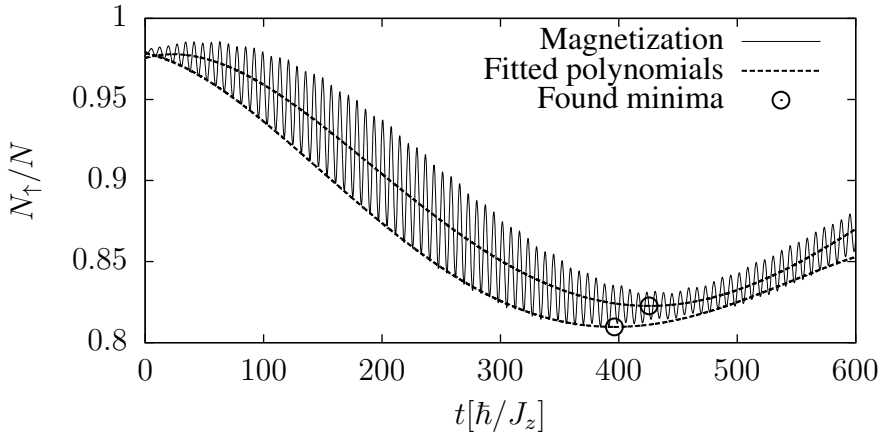


Figure 6.8: Magnetization in the presence of a laser with frequency near resonance with the $n = 2$ excitation. Polynomials are fitted to the signal and the lower edge of the signal.

6.3.1 Individual resonances

Having confirmed that the system has resonance near the expected energies, the next step was to consider each resonance in turn. We turned the laser amplitude down to a smaller, more realistic value of $0.01J_z$ and ran the simulation with laser frequencies near each of the minima we discovered above.

A typical run near resonance with the $n = 2$ excitation is shown in figure 6.8. The magnetization oscillates quickly while falling off to a minimum. The exact position of the minimum is disturbed by the oscillations, so to find as precise an estimate of the Rabi frequency as possible we tried to remove the effect of the oscillations by fitting a polynomial to the signal. As another option, we found the lower edge of the rapid oscillations by detecting a sign change of the derivative, and fitted a polynomial to this lower edge. The two polynomials are shown in the plot, along with their minimal values.

The time t_{min} until the minimal value is reached is half a Rabi oscillation, so the Rabi frequency is calculated as $2\pi/2t_{\text{min}} = \pi/t_{\text{min}}$. From section 3.3, we expect that the Rabi frequency depends on the laser frequency as

$$\omega_R^2 = \delta^2 + k^2 = \omega^2 - 2\omega\omega_0 + \omega_0^2 + V_0^2 |\langle g| \hat{S}^z |e\rangle|^2, \tag{6.9}$$

so that plotting ω_R^2 against the laser frequency should give a parabola with a minimum at the resonance frequency $\omega = \omega_0$. The squared estimated Rabi frequencies for laser frequencies near the $n = 2$ resonance are plotted in figure 6.9, with fitted parabolas. We do find an approximately parabolic behavior of ω_R^2, but the quadratic coefficient is significantly less than 1 (about 0.6), meaning that the parabola is shallower than expected.

Having found a good resonance frequency we did another run at resonance with a higher value of χ_{max}. Comparing the two runs shows similar magnetization behaviours, providing confidence that the results are correct. To get another indication of whether the laser frequency is close to resonance we calculated the overlap between the ground state $|\psi_0\rangle$ and a state $|\psi_R\rangle$ saved at a time near t_{min}. The overlap had a squared magnitude of $4 \cdot 10^{-4}$, which indicates that

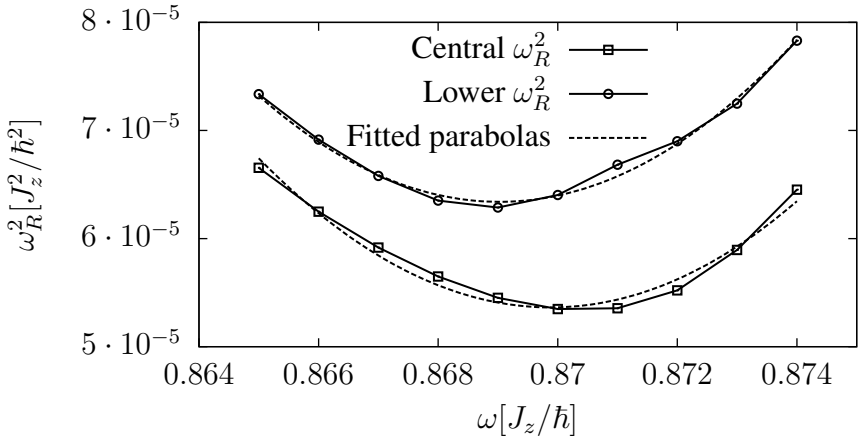


Figure 6.9: Two estimates for the squared Rabi frequency as a function of laser frequency near the $n = 2$ resonance, using polynomials fitted to the signal (central ω_R) and to the lower edge of the signal (lower ω_R). A parabola is fitted to both sets of estimates.

Table 6.1: Estimated resonance values for the excited states. The data for $n = 8$ was not reliable enough for a quadratic fit.

n	ω_0	Quadratic coefficient	$\mid \langle \psi_R \mid \psi_0 \rangle \mid^2$
2	0.869	0.6	$4 \cdot 10^{-3}$
4	1.091	1.02	$6 \cdot 10^{-4}$
6	1.253	0.86	$2 \cdot 10^{-3}$
8	1.385	N/A	0.4
10	1.494	1.04	$5 \cdot 10^{-2}$
12	1.593	0.81	0.2

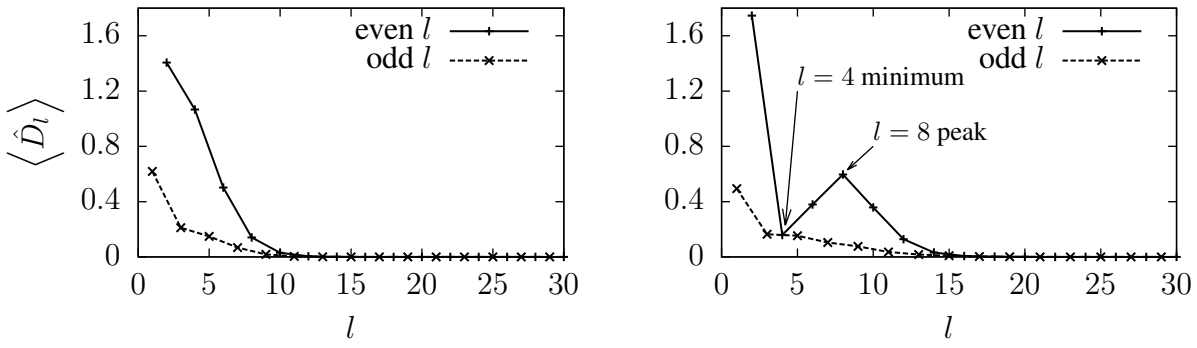
the resonance is very good.

Similar analyses were performed for all the excitations up to $n = 12$. Above this the coupling to the ground state was too weak to find resonances with our method. The estimated resonance excitation energies $\hbar\omega_0$ are listed in table 6.1, along with the quadratic coefficients of the parabolic fits and the overlaps with the ground state. For the $n = 8$ excitation the truncation errors were too big to get reliable Rabi frequencies for all the laser frequencies. This means that the given resonance value is uncertain, and indeed the overlap with the ground state is large in this case. The large overlap at $n = 12$ indicates that a better resonance frequency can be found by a search with higher resolution in ω.

6.3.2 Domain distributions

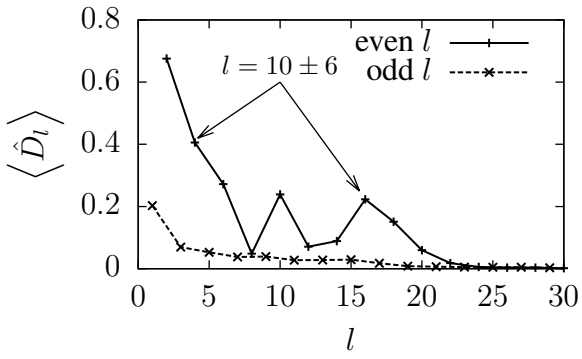
The resonance states $|\psi_R\rangle$, having small overlaps with the ground state, are expected to resemble the single-domain wavefunctions $|n, l\rangle$ from section 3.2.4. To investigate this we calculated the domain distributions of the $|\psi_R\rangle$ states (see section 6.2.1). The result is shown for some of the resonances in figure 6.10.

Comparing with the wavefunctions from figure 3.6, we see broadly similar features, such as a minimum at $l = n = 4$ and peaks at $l = n, l = n \pm 6$ for the higher n states. The peak at

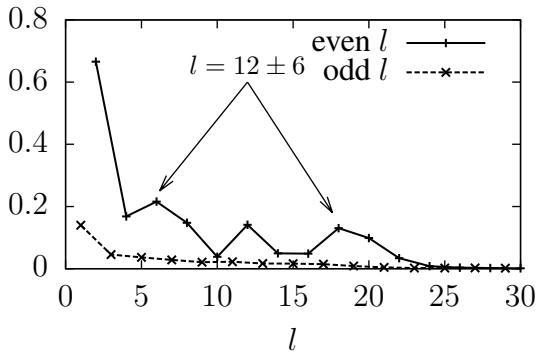


(a) $n = 2$

(b) $n = 4$

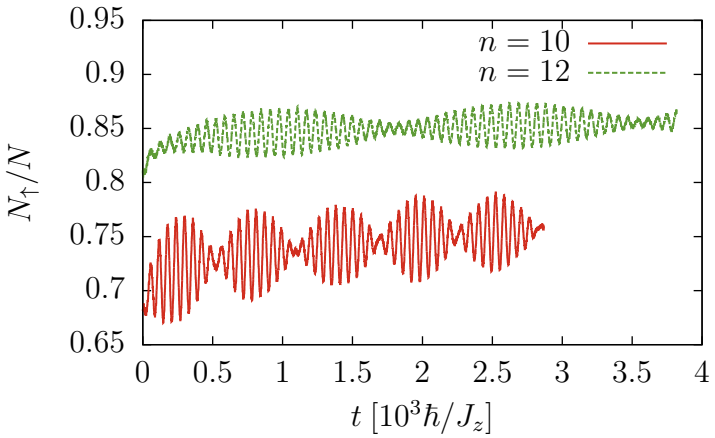


(c) $n = 10$

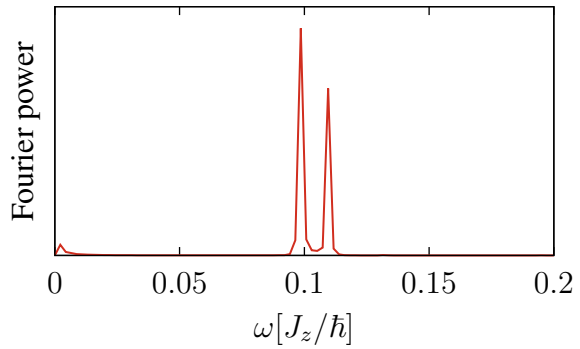


(d) $n = 12$

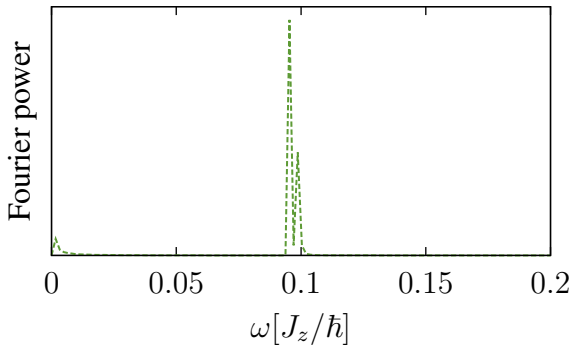
Figure 6.10: Domain distributions for the states at resonance with the indicated excitations. We recognize some features of the theoretical wavefunctions: A minimum $l = n = 4$, and peaks at $l = n$ and $l = n \pm 6$ for high n, although the $n = 10,\ l = 4$ peak seems to be washed out by high values at low l.



(a) Magnetization after half a Rabi oscillation to the indicated resonance



(b) Fourier spectrum of the $n = 10$ Bloch oscillation



(c) Fourier spectrum of the $n = 12$ Bloch oscillation

Figure 6.11: Bloch oscillations obtained by tuning a strong laser to the $n = 10$ or $n = 12$ resonances, with amplitude $0.1J_z$. We see a beat pattern due to the slight deviations from the Wannier-Zeeman ladder, as confirmed by the Fourier spectra.

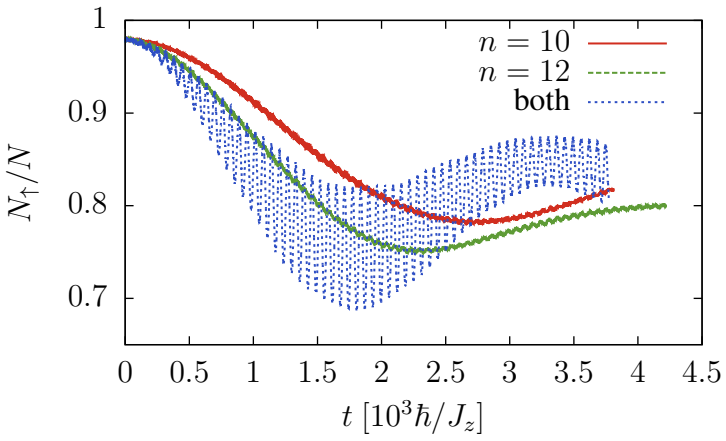
$n = 10, l = 4$ seems to be missing. However, the value at $l = 4$ is still larger when $n = 10$ than when $n = 12$.

We can not hope to find any quantitative agreement, as the value of $\langle D_l \rangle$ is an extensive quantity due to the possibility of several domains in a single state, while the wavefunction is intensive. Correcting for this effect in a precise way requires a theory of how domains interact, which is beyond the scope of this thesis.

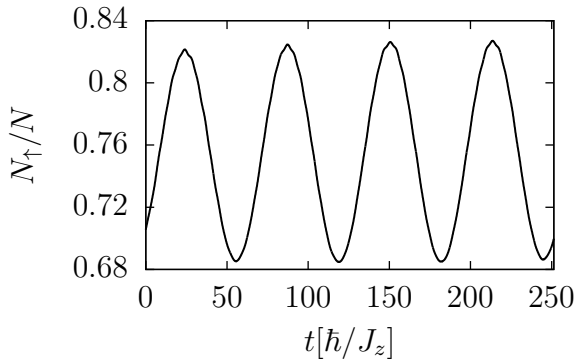
6.4 Bloch oscillations

Having found reliable resonances we now make use of the precise resonance frequencies to excite a Bloch oscillation. To that end we need to populate two of the excitations we have been targeting (see section 3.1.3). Shinkevich and Syljuåsen (2013) suggested to use a high laser amplitude induce get off-resonant excitation of the neighboring $n \pm 2$ states in addition to populating the resonant state.

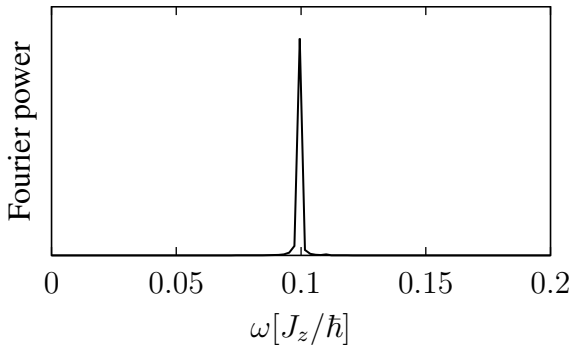
We did two runs with a laser amplitude of $0.1h_z$, targeting the $n = 10$ and $n = 12$ resonances, resulting in Bloch oscillations after the laser was turned off. These oscillations are



(a) Magnetization when tuning lasers to the indicated resonances.



(b) Bloch oscillations after turning off the laser.



(c) Fourier spectrum of the Bloch oscillation.

Figure 6.12: Bloch oscillations obtained by tuning two lasers the resonance frequencies found for $n = 10$ and $n = 12$, with amplitudes $0.01J_z$ and $0.03J_z$. Rabi oscillations between the ground state and a Bloch oscillating mode are induced by the lasers. Turning off the laser at $t = 1750$ yields the oscillations in (b).

shown in figure 6.11, where we see a characteristic beat pattern indicating *two* oscillations with slightly different frequencies. This is due to the deviations from the Wannier-Zeeman ladder at low n, so that $E_n - E_{n-2}$ is slightly different from $E_{n+2} - E_n$. The beat is faster for $n = 10$ than $n = 12$, as the differences are larger for smaller n.

While this is a simple method for exciting Bloch oscillations, it requires a very high laser amplitude to get significant off-resonant excitation given the small couplings. A laser amplitude of $0.1J_z$ requires electric fields an order of magnitude larger than the typical dielectric breakdown limit for an insulator (Shinkevich and Syljuåsen, 2013).

In order to excite two energy levels with a smaller laser amplitude, one could use two lasers, each in resonance with an energy level. For this to work well it is necessary that the Rabi oscillations to the two excitations roughly coincide, so that we can turn off the laser at a point where both energy levels are excited. Since the coupling to the higher energy level is weaker, this requires a higher laser amplitude for this excitation.

As we can see in figure 6.12, tuning one laser to the $n = 10$ resonance with amplitude $0.01J_z$ creates a Rabi oscillation with about the same frequency as a laser at the $n = 12$ resonance

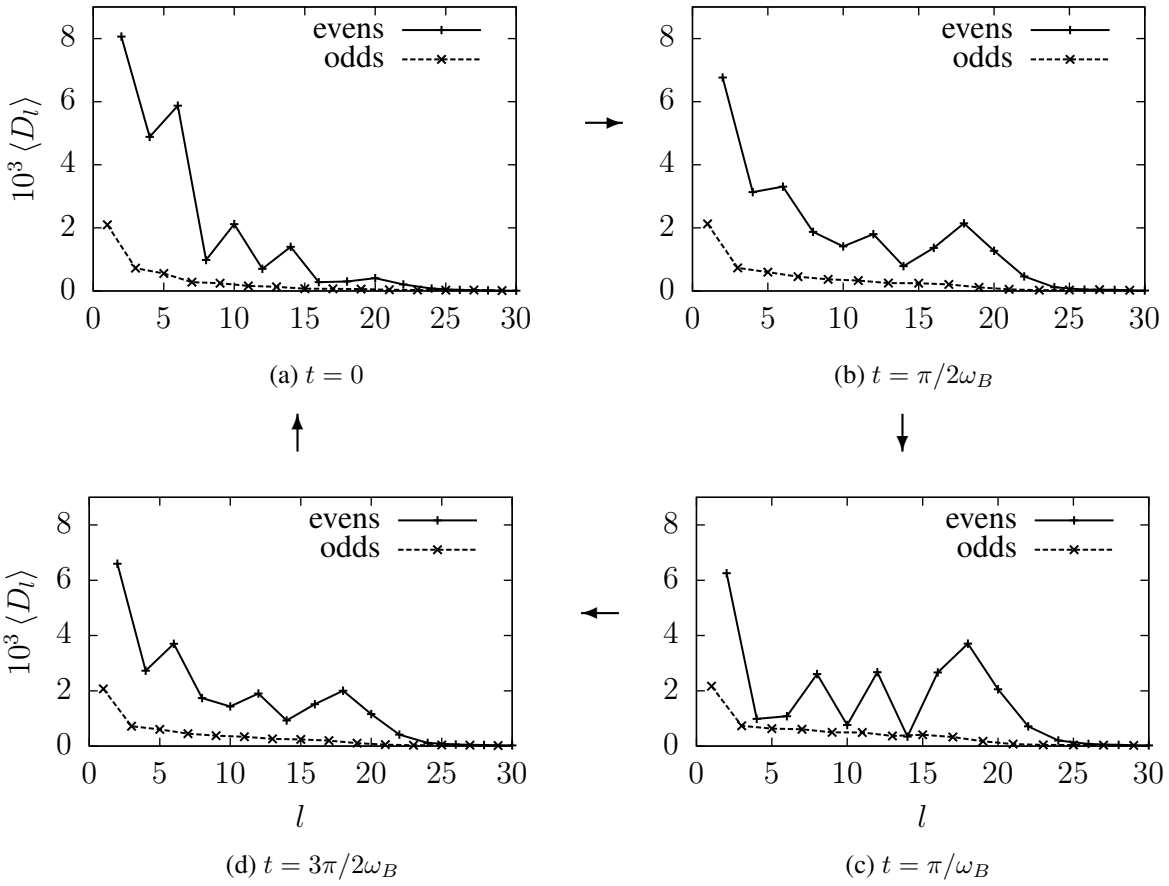


Figure 6.13: Domain distributions during one Bloch oscillation, showing transitions between states resembling the $n = 10$ and $n = 12$ states in figure 6.10.

with amplitude $0.03J_z$. Running both lasers at the same with these parameters creates a Rabi oscillation between a Bloch oscillating mode and a non-oscillating mode. Turning off the laser at $t = 1750\hbar/J_Z$ yields the Bloch oscillations seen in figure 6.12b. These oscillations were very stable for at least 50 periods, at which point we stopped the simulation.

It is interesting to take a closer look at how the state evolves during a Bloch oscillation. Since we have excited two different eigenstates, we should expect to see an oscillation between the domain distributions of the two states. We saved snapshots of the state during one Bloch oscillation and calculated the domain distributions for these states. Four such distributions are shown in figure 6.13. The distribution in (a), which occurs at maximal magnetization, shows peaks at $l = 10, 16$, characteristic of the $n = 10$ state. Similarly, the distribution in (c), which occurs at maximum magnetization, shows peaks characteristic of the $n = 12$ state. The remaining distributions in (b) and (d) interpolate between this. One puzzling observation is that the peaks at $l = n - 6$ seem to be missing, replaced with peaks at $l = n - 4$.

One of the interesting properties of Bloch oscillations is that the frequency ω_B of the oscillation is tunable by the external field strength, by $\omega_B = 2h_z/\hbar$. It is therefore interesting to check whether this is still the case here. Starting anew from a different magnetic field would be very time-consuming (involving a new search for resonance frequencies), but we can start from

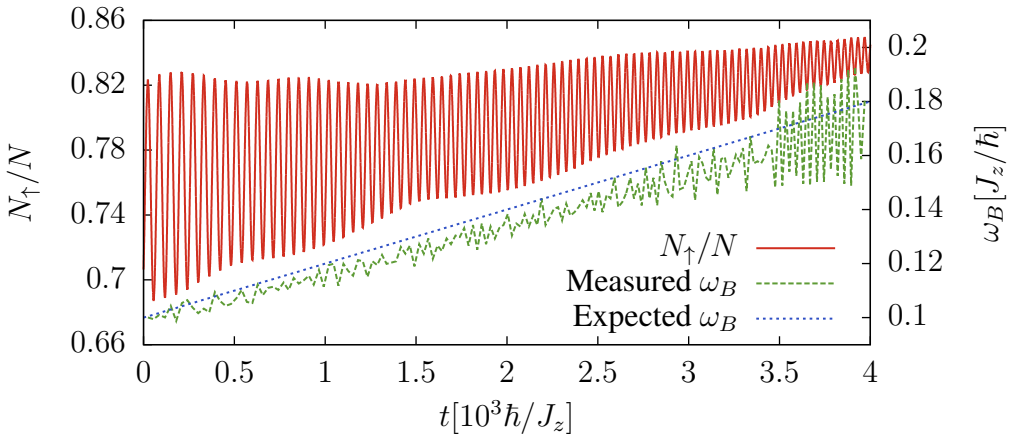


Figure 6.14: Time evolution of the Bloch oscillating state while slowly increasing the external field by $h_z = (0.05 + 10^{-5}tJ_z/\hbar)J_z$. The frequency is measured by computing the time differences between the maxima and minima, and plotted along with the expected Bloch frequency.

the Bloch oscillating state we have found and slowly change the magnetic field. If the change is slow enough, the state should adiabatically adjust to the field and continue to oscillate with the adjusted Bloch frequency.

Such a numerical experiment is shown in figure 6.14, where we increase the field by $h_z = (0.05 + 10^{-5}tJ_z/\hbar)J_z$. The magnetization is seen to oscillate with some frequency, and one can barely make out a higher frequency at late times. By finding the local minima and maxima of the magnetization one can estimate the frequency. This estimate is also plotted along with the expected Bloch frequency. The measured frequency agrees with the expected Bloch frequency up to some amount of noise, which is probably due to inaccuracies in the measured positions of the minima and maxima.

6.5 Discussion

We succeeded in finding Bloch oscillations by targeting specific excitation energies in the Wannier-Zeeman ladder. The oscillations seem robust once excited, which is a nontrivial result in this interacting quantum many-body problem.

The amplitudes of the Rabi oscillations were smaller than expected. If one achieves a good number of domains by Rabi oscillations one should expect a magnetization approaching $N_\uparrow/N \sim 0.5$, but our results fall short of this. In addition the Rabi frequencies at resonance are higher than those found in Shinkevich and Syljuåsen (2013), which indicates that interactions between domains play an important part and lead to blocking (see section 3.3). An alternative explanation might be that finite-size effects prevent some of the domains from forming. However, repeats of our crucial runs at $N = 200$ give similar values.

The process of exciting Bloch oscillations seems to require significant fine-tuning of the laser frequency. At a laser amplitude of $0.01J_z$ the frequency must be correct to a relative accuracy of at least $1/1000$, even for the lowest-lying excitations with strong couplings to the ground state. At zero momentum the approximately equally-spaced Wannier-Zeeman ladder

begins at $n = 12$, but even the $n = 14$ state coupled too weakly to the ground state for us to find it with the unrealistically high laser amplitude of $0.1J_z$. The laser amplitude used here is also very large: Even the low amplitude of $0.01J_z$ requires a magnetic field amplitude of 0.15T, which must be pulsed somewhat accurately for $2000\hbar/J_z \sim 0.5$ns. Lower laser amplitudes could be used. However, this will require even more fine-tuning of the frequency, and the system will need to stay coherent during the longer Rabi period in order to show Bloch oscillations.

The method of exciting Bloch oscillations with lasers seems somewhat suboptimal, as we target only a few rungs of the Wannier-Zeeman ladder. By targeting more excitations simultaneously, one might achieve Bloch oscillations more easily. This could be achieved by more violent perturbations of the crystal, such as quick rotations of the magnetic field or rapid changes of field strength. We attempted to simulate some experiments of this kind. However, it turns out that such violent perturbations lead to rapid growth of entanglement in the short term, which means that χ_{max} needs to be very big in order to simulate the system accurately. This again means that the simulation is too computationally demanding to look for phenomena at the timescales of Bloch oscillations. Therefore no clear results came out of these numerical experiments.

Chapter 7

Conclusion

We have seen that crystals of $CoCl_2 \cdot 2\,H_2O$, modeled as a one-dimensional ferromagnetic chain, should exhibit magnetic Bloch oscillations when subjected to a magnetic field. The quantum signature of Bloch oscillations is a ladder of equally spaced energy levels, dubbed the Wannier-Zeeman ladder. Any linear combination of states from this ladder will cause the magnetization to oscillate with a characteristic frequency determined by the magnetic field strength.

One possible method of observing the oscillations is to use lasers in resonance with the energies of the ladder to induce Rabi oscillations to these states. Simulating this process in the presence of interactions between domains requires many-body quantum mechanical calculations on a one-dimensional chain, which the TEBD method is ideally suited for.

We developed an implementation of the TEBD method for simulating the above system. Verification checks confirm that it is working correctly. The energies of the Wannier-Zeeman ladder can't be known precisely, so we conducted a search with different laser frequencies to find the best possible resonances. Once these were found, lasers could be tuned to provoke Bloch oscillations, which were observed and found to be robust. This confirms that Bloch oscillations in our model survive any decoherence that interactions between domains might lead to. We confirmed that the oscillation frequency is tunable by the external field.

References

Bloch, F. (1929). Über die Quantenmechanik der Elektronen in Kristallgittern. *Zeitschrift für Physik*, 52(7-8):555–600.

Christensen, N., Lefmann, K., Johannsen, I., and Jørgensen, O. (2000). Magnetic bloch oscillations in the near-ising antiferromagnet cocl2·2d2o. *Physica B: Condensed Matter*, 276–278(0):784 – 785.

Feldmann, J., Leo, K., Shah, J., Miller, D. A. B., Cunningham, J. E., Meier, T., von Plessen, G., Schulze, A., Thomas, P., and Schmitt-Rink, S. (1992). Optical investigation of Bloch oscillations in a semiconductor superlattice. *Physical Review B*, 46:7252–7255.

Kittel, C. (2005). *Introduction to Solid State Physics*. Wiley, eigth edition.

Krieger, J. B. and Iafrate, G. J. (1986). Time evolution of Bloch electrons in a homogeneous electric field. *Physical Review B*, 33:5494–5500.

Kyriakidis, J. and Loss, D. (1998). Bloch oscillations of magnetic solitons in anisotropic spin-$\frac{1}{2}$ chains. *Physical Review B*, 58:5568–5583.

Lay, D. C. (2006). *Linear Algebra and its Applications*. Addison Wesley, third edition.

Leo, K., Bolivar, P. H., Brüggemann, F., Schwedler, R., and Köhler, K. (1992). Observation of Bloch oscillations in a semiconductor superlattice. *Solid State Communications*, 84(10):943 – 946.

Montfrooij, W., Granroth, G. E., Mandrus, D. G., and Nagler, S. E. (2001). Spin dynamics of the quasi-one-dimensional ferromagnet $CoCl_2 \cdot 2\,D_2O$. *Physical Review B*, 64:134426.

Sakurai, J. J. (1993). *Modern Quantum Mechanics (Revised Edition)*. Addison Wesley, 1 edition.

Sanderson, C. (2013). Armadillo: C++ linear algebra library. `http://arma.sourceforge.net/`. Accessed December 1, 2013.

Shinkevich, S. and Syljuåsen, O. F. (2012). Spectral signatures of magnetic Bloch oscillations in one-dimensional easy-axis ferromagnets. *Physical Review B*, 85:104408.

Shinkevich, S. and Syljuåsen, O. F. (2013). Numerical simulations of laser-excited magnetic Bloch oscillations. *Physical Review B*, 87:060401.

Sornborger, A. T. and Stewart, E. D. (1999). Higher-order methods for simulations on quantum computers. *Physical Review A*, 60:1956–1965.

Suzuki, M. (1990). Fractal decomposition of exponential operators with applications to many-body theories and monte carlo simulations. *Physics Letters A*, 146(6):319 – 323.

Torrance, J. B. and Tinkham, M. (1969). Excitation of multiple-magnon bound states in $CoCl_2 \cdot 2H_2O$. *Physical Review*, 187:595–606.

Verstraete, F. and Cirac, J. I. (2006). Matrix product states represent ground states faithfully. *Physical Review B*, 73:094423.

Vidal, G. (2003). Efficient classical simulation of slightly entangled quantum computations. *Physical Review Letters*, 91:147902.

Wannier, G. H. (1960). Wave functions and effective hamiltonian for Bloch electrons in an electric field. *Physical Review*, 117:432–439.

Waschke, C., Roskos, H. G., Schwedler, R., Leo, K., Kurz, H., and Köhler, K. (1993). Coherent submillimeter-wave emission from Bloch oscillations in a semiconductor superlattice. *Physical Review Letters*, 70:3319–3322.